

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UM MODELO DE ANÁLISE DE
VIABILIDADE PARA DESENVOLVIMENTO
DE COMPONENTES**

PETER SCHUCK HEMESATH

Orientador: Prof. Dr. Ricardo Melo Bastos

Dissertação de Mestrado

PORTO ALEGRE
2006

PETER SCHUCK HEMESATH

**UM MODELO DE ANÁLISE DE VIABILIDADE PARA
DESENVOLVIMENTO DE COMPONENTES**

Dissertação apresentada como requisito parcial à
obtenção do grau de Mestre em Ciência da
Computação, pelo Programa de Pós-Graduação em
Ciência da Computação, da Pontifícia Universidade
Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Ricardo Melo Bastos.

PORTO ALEGRE
2006

Dados Internacionais de Catalogação na Publicação (CIP)

H488u Hemesath, Peter Schuck.

Um modelo de análise de viabilidade para desenvolvimento de componentes / Peter Schuck Hemesath. – Porto Alegre, 2006.
95 f.

Diss. (Mestrado) – Fac. de Informática, PUCRS.
Orientador: Profa. Dr. Ricardo Melo Bastos

1. Informática. 2. Engenharia de Software. 3.

Desenvolvimento Baseado em Componentes. I. Bastos,

Ricardo Melo. II. Título.

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada “*Um Modelo de Análise de Viabilidade para Desenvolvimento de Componentes*”, apresentada por Peter Schuck Hemesath, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas de Informação, aprovada em 23/03/2006 pela Comissão Examinadora:



Prof. Dr. Ricardo Melo Bastos – PPGCC/PUCRS
Orientador



Profa. Dra. Karin Becker – PPGCC/PUCRS



Prof. Dr. Toacy Cavalcante de Oliveira – PPGCC/PUCRS



Prof. Dr. Sergio Crespo Coelho da Silva Pinto – UNISINOS

Homologada em 29/07/09, conforme Ata No. 17/09 pela Comissão Coordenadora.



Prof. Dr. Fernando Luís Dotti
Coordenador.

AGRADECIMENTOS

Aos meus pais, Rolf e Magda pelo exemplo de vida, dedicação e amor. Obrigado por sempre me incentivarem na busca pelo conhecimento. Esta conquista também é de vocês!

À minha família pelo incentivo constante, o carinho e a compreensão nos momentos difíceis, a palavra amiga procurando amenizar minha ansiedade diante dos obstáculos. Cada um de vocês contribuiu da sua maneira para esta vitória.

Ao meu orientador, Ricardo M. Bastos por sua dedicação, idéias e sobretudo a sua tranqüilidade na condução deste projeto. Muito Obrigado!

Ao professor Marcelo Blois pelo apoio incondicional durante a fase mais difícil do mestrado e pelas suas idéias que ajudaram a conduzir este trabalho ao seu estado atual.

Ao professor Toacy Cavalcanti pelo acompanhamento do trabalho com sugestões e críticas, sempre procurando contribuir.

Às empresas participantes da pesquisa de campo por permitir a sua realização e a todos os profissionais que dedicaram o seu tempo para contribuir com este trabalho.

Aos Convênios Dell/PUCRS e Compasso/PUCRS pelo apoio financeiro para realização deste trabalho.

Ao corpo docente e funcionários da FACIN/PPGCC pelo suporte oferecido.

RESUMO

O acelerado crescimento e a competição pelo mercado têm levado empresas a buscarem formas de continuamente aumentar a qualidade e produtividade nos seus processos de desenvolvimento de software. Uma forma para se otimizar o processo de desenvolvimento de software é a adoção de técnicas de reuso de software como Desenvolvimento Baseado em Componentes (DBC) que abrange preocupações como o aumento da produtividade e a redução dos custos com qualidade. Mas será o desenvolvimento baseado em componentes, por si só, garantia de benefício econômico, ou seja, os custos de compra/integração de componentes reutilizáveis serão sempre menores do que o custo de desenvolvimento das mesmas funcionalidades para uma dada aplicação? E o desenvolvimento de um componente reutilizável será garantia de recuperação do investimento realizado para criação do mesmo?

Neste contexto, o presente trabalho apresenta uma proposta de modelo de análise de viabilidade para a criação de componentes, que com base em uma série de entradas qualitativas e quantitativas, provê projeções de retorno sobre o investimento na criação de componentes. Tal modelo tem como propósito servir como ferramenta de auxílio à tomada de decisão para os engenheiros de software envolvidos em programas de desenvolvimento de componentes reutilizáveis.

Palavras-chave: Desenvolvimento Baseado em Componentes. Métricas. Armazenamento. Retorno de investimento. Criação de componentes.

ABSTRACT

The speed up growth and the market competition have taken companies to search for different ways to continuously increase quality and productivity in its software development processes. One way to optimize a software development process is the adoption of software reuse techniques, such as Component Based Development (CBD), which concerns productivity increase and cost reduction with quality. But it will be the component based development, by itself, a guarantee of economic benefit, or either, the costs to buy or integrate those components will always be lesser than the development cost of the same functionalities for one given application? Will the development of reusable components be a guarantee for recovering the accomplished investment from its creation? In this context, this work presents a proposal for a viability analysis model of reusable components creation, based on a series of qualitative and quantitative inputs, which can provide investment return projections for the creation of reusable components. Such model intends to serve as an assistance tool to the software engineers decision making process involved in reusable components development.

Keywords: Component-Based Development. Metrics. Storage. Return on Investment. Creating Components.

LISTA DE FIGURAS

Figura 1. Desenho de Pesquisa	16
Figura 2. Processo de DBC, adaptado de [CRN02].....	22
Figura 3. Modelo espiral para DBC, adaptado de [SAM97].....	23
Figura 4. Modelo do processo de DBC, adaptado de [BER00].....	24
Figura 5. Abordagem GQM para métricas de DBC, adaptado de [DAG03]	30
Figura 6. Retorno do investimento em componentização	52
Figura 7. Cenário de reuso de componentes.....	67
Figura 8. Ferramenta de aplicação do modelo (dados do componente).....	69
Figura 9. Ferramenta de aplicação do modelo (dados das aplicações)	71
Figura 10. Ferramenta de aplicação do modelo (dados gerais)	73
Figura 11. Ferramenta de aplicação do modelo (simulação de retorno de investimento)	74
Figura 12. Ferramenta de aplicação do modelo (histórico de simulações)	75

LISTA DE TABELAS

Tabela 1. Métricas utilizadas na composição do modelo.....	41
Tabela 2. Fatores que influenciam o reuso de componentes.....	44
Tabela 3. Dados dos Componentes.....	57
Tabela 4. Dados históricos.....	58
Tabela 5. Passos para cálculo do Payback do componente A.....	58
Tabela 6. Análise de retorno de investimento com Payback.....	59
Tabela 7. Análise de retorno de investimento com VPL.....	61
Tabela 8. Pesos dos fatores de ajuste.....	64
Tabela 9. Riscos do componente A.....	64
Tabela 10. Riscos do componente B.....	65
Tabela 11. Riscos do componente C.....	65
Tabela 12. Cálculo da probabilidade de reuso do componente A.....	66
Tabela 13. Análise de retorno de investimento com fatores qualitativos.....	66
Tabela 14. Comparativo das propostas de modelos.....	83

LISTA DE ABREVIATURAS

DBC	<i>Desenvolvimento Baseado em Componentes</i>
CDPe	<i>Centro de Desenvolvimento e Pesquisa em e-Business</i>
CLOC	<i>Linhas de Código Comentadas</i>
FP	<i>Pontos de Função</i>
LOC	<i>Linhas de Código</i>
NLOC	<i>Linhas de Código Não Comentadas</i>
PUCRS	<i>Pontifícia Universidade Católica do Rio Grande do Sul</i>
UM	<i>Unidades de Medida</i>
UML	<i>Unified Modeling Language</i>
VPL	<i>Valor Presente Líquido</i>

SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	ETAPAS DA PESQUISA	14
1.2	ESTRUTURA DO DOCUMENTO	17
2	DESENVOLVIMENTO BASEADO EM COMPONENTES.....	18
2.1	DEFINIÇÃO DE COMPONENTES.....	18
2.2	O PROCESSO DE DESENVOLVIMENTO BASEADO EM COMPONENTES	19
2.2.1	Processos e Atividades para DBC Encontrados na Literatura.....	20
2.2.2	Abordagem de Ivica Crnkovic [CRN02].....	21
2.2.3	Abordagem de Johannes Sametinger [SAM97].....	22
2.2.4	Abordagem de Klaus Bergner.....	23
2.3	CONSIDERAÇÕES.....	25
3	MÉTRICAS E RISCOS PARA O DESENVOLVIMENTO BASEADO EM COMPONENTES	26
3.1	MÉTRICAS PARA O DESENVOLVIMENTO BASEADO EM COMPONENTES	26
3.1.1	Abordagem de Aldo Dagnino et. al. [DAG03].....	27
3.1.2	Abordagem de Jeffrey Poulin e Joseph Caruso [POU93]	36
3.1.2.1	<i>Custo poupado pelo reuso</i>	36
3.1.2.2	<i>Custo de desenvolver componente</i>	38
3.1.3	Considerações	40
3.2	RISCOS PARA O DESENVOLVIMENTO BASEADO EM COMPONENTES	41
3.2.1	Considerações	43
4	ANÁLISE FINANCEIRA PARA CRIAÇÃO DE COMPONENTES.....	45
4.1	PAYBACK	45
4.2	VALOR PRESENTE LÍQUIDO	47
4.2.1	Taxa Mínima de Atratividade	48
4.3	CONSIDERAÇÕES.....	48
5	MODELOS PARA ANÁLISE DE RETORNO DE INVESTIMENTO.....	49
5.1	ABORDAGEM DE Jeffrey Poulin e Joseph Caruso [POU93]	49
5.2	ABORDAGEM DE RUTH MALAN E KEVIN WENTZEL [MAL93].....	50
6	MODELO PROPOSTO.....	51
6.1	ANÁLISE QUANTITATIVA BÁSICA (PAYBACK)	53
6.1.1	Custo de Desenvolvimento do Componente (CDC)	54
6.1.2	Custo de Reuso do Componente (CRC).....	54
6.1.3	Custo de não Reutilizar o Componente (CNR)	55
6.1.3.1	<i>Custo de Desenvolvimento</i>	56
6.1.3.2	<i>Custo de Manutenção</i>	56
6.1.4	Aplicação do Modelo	57
6.2	ANÁLISE QUANTITATIVA COMPOSTA (<i>VALOR PRESENTE LÍQUIDO</i>).....	59
6.2.1	Aplicação do Modelo	60

6.3	ANÁLISE QUALITATIVA	62
6.3.1	Quantificação de Valores Qualitativos.....	63
6.3.2	Aplicação do Modelo	64
6.3.3	Automatização do Modelo.....	68
7	AVALIAÇÃO DO MODELO.....	76
7.1	CARACTERIZAÇÃO DAS ORGANIZAÇÕES	76
7.2	ROTEIRO	76
7.3	CARACTERIZAÇÃO DOS RESPONDENTES.....	77
7.4	RESULTADOS.....	77
8	CONSIDERAÇÕES FINAIS	82
8.1	TRABALHOS RELACIONADOS FRENTE AO MODELO PROPOSTO.....	82
8.2	CONTRIBUIÇÕES	83
8.3	TRABALHOS FUTUROS	84
	REFERÊNCIAS BIBLIOGRÁFICAS	85
	APÊNDICE I - INSTRUMENTO DE COLETA DE DADOS DA PESQUISA DE	
	CAMPO	90
	APÊNDICE II – RELAÇÃO DAS EQUAÇÕES APRESENTADAS	94

1 INTRODUÇÃO

O acelerado crescimento e a competição pelo mercado têm levado empresas a buscarem formas de continuamente aumentar a qualidade e produtividade nos seus processos de desenvolvimento de software. Uma forma para otimizar o processo de desenvolvimento de software é a adoção de técnicas de reuso de software como Desenvolvimento Baseado em Componentes (DBC) que abrange preocupações como o aumento da produtividade e a redução dos custos com qualidade.

Conforme [SAM97] e [CRN02], o desenvolvimento baseado em componentes apresenta duas perspectivas, o desenvolvimento com componentes e o desenvolvimento de componentes. O desenvolvimento com componentes foca no desenvolvimento de aplicações a partir da composição de artefatos reutilizáveis, partindo do princípio que existam componentes disponíveis e que supram, mesmo que parcialmente, os requisitos da aplicação. O desenvolvimento de componentes envolve preocupações de como gerar artefatos reutilizáveis. Desta forma, o reuso de componentes pode ser visto como um modelo econômico de oferta e demanda, respectivamente representados por produtores e consumidores de componentes.

Mas será o desenvolvimento baseado em componentes, por si só, garantia de benefício econômico, ou seja, os custos de compra/integração de componentes reutilizáveis serão sempre menores do que o custo de desenvolvimento das mesmas funcionalidades para uma dada aplicação? E o desenvolvimento de um componente reutilizável será garantia de recuperação do investimento realizado para criação do mesmo? A literatura aponta que a resposta para estas perguntas é “não” [SAM97], [CRN02], [SZY99]. Diversos fatores podem afetar o benefício econômico relacionado ao DBC, sucintamente podemos citar:

- altos custos na integração do componente devido a necessidade da realização de adaptações para o seu reuso podem elevar o custo de reutilização;
- falta de documentação adequada pode impossibilitar o reuso do componente;
- o componente pode se tornar obsoleto antes de retornar o investimento realizado na sua criação.

Para responder a essas perguntas é necessária a criação de modelos de análise de retorno de investimento para DBC que possam auxiliar o engenheiro de software na tomada de decisão de reuso ou criação de componentes. Um estudo sobre métricas para DBC, realizado dentro do âmbito da pesquisa, apontou uma carência de propostas para análise de retorno de investimento.

Em um âmbito financeiro, a criação de componentes pode ser vista como um investimento, e como todo investimento, é esperado que o mesmo seja financeiramente rentável. Componentes retornam o investimento referente a sua criação através do seu reuso em múltiplos projetos. Rentabilidade com componentes é obtida pela redução do esforço no desenvolvimento do software em função da integração do componente, frente à implementação de todas as funcionalidades (encapsuladas no componente) para cada projeto.

Engenheiros de software precisam estar aptos para traduzir os benefícios de reuso em benefícios contábeis para que o investimento em reuso possa ser comparado a outras alternativas de investimento corporativas [PFL96]. Nesse contexto, o presente trabalho apresenta uma proposta de modelo de análise de viabilidade para a criação de componentes que, com base em uma série de entradas qualitativas e quantitativas, provê projeções de retorno sobre o investimento na criação de componentes. Tal modelo tem como propósito servir como ferramenta de auxílio à tomada de decisão para os engenheiros de software envolvidos em programas de desenvolvimento de componentes reutilizáveis. A pesquisa que levou a construção do modelo foi financiada através de dois convênios com fins de incentivo à pesquisa. Inicialmente no Centro de Desenvolvimento e Pesquisa em e-Business (CDPe), do convênio Dell – PUCRS e posteriormente na empresa Compasso, que também se tornou possível através de um convênio desta empresa com a PUCRS.

1.1 ETAPAS DA PESQUISA

O trabalho apresentado neste documento foi realizado através de etapas, as quais estão representadas na Figura 1. As etapas estão representadas através de um diagrama de atividades UML (*Unified Modeling Language*) [UML03].

Primeiramente foi realizado o levantamento bibliográfico e estudos iniciais sobre o tema. Esses estudos permitiram identificar as carências relativas à área, as quais serviram de motivação para este trabalho.

Para criação do modelo de análise de viabilidade foram identificadas quais as principais fontes de custos referentes ao DBC e quais as métricas que contabilizavam os custos das atividades do processo de desenvolvimento de componentes. Dessa forma foi estabelecida a parte dita quantitativa do modelo. Para construir um modelo que permitisse uma análise de retorno de investimento mais precisa foi integrada ao mesmo uma série de fatores qualitativos. Esses fatores permitem ao modelo a possibilidade de contabilizar as reduções de lucro, referentes ao decréscimo da reusabilidade do componente, propiciadas pela obsolescência do artefato no decorrer do tempo.

Para auxiliar a aplicação do modelo foi desenvolvida uma ferramenta cujo objetivo é de facilitar a aplicação do mesmo, propiciando formas de análise visual do retorno sobre investimento e a comparação de múltiplos cenários para desenvolvimento de componentes. Partindo do modelo de análise de viabilidade proposto, foi realizada uma análise de campo para observar a aplicabilidade do modelo na tomada de decisão sob o ponto de vista de profissionais de duas organizações de desenvolvimento de software. Com base no retorno desses profissionais o modelo pôde ser avaliado e evoluído, para a presente forma.

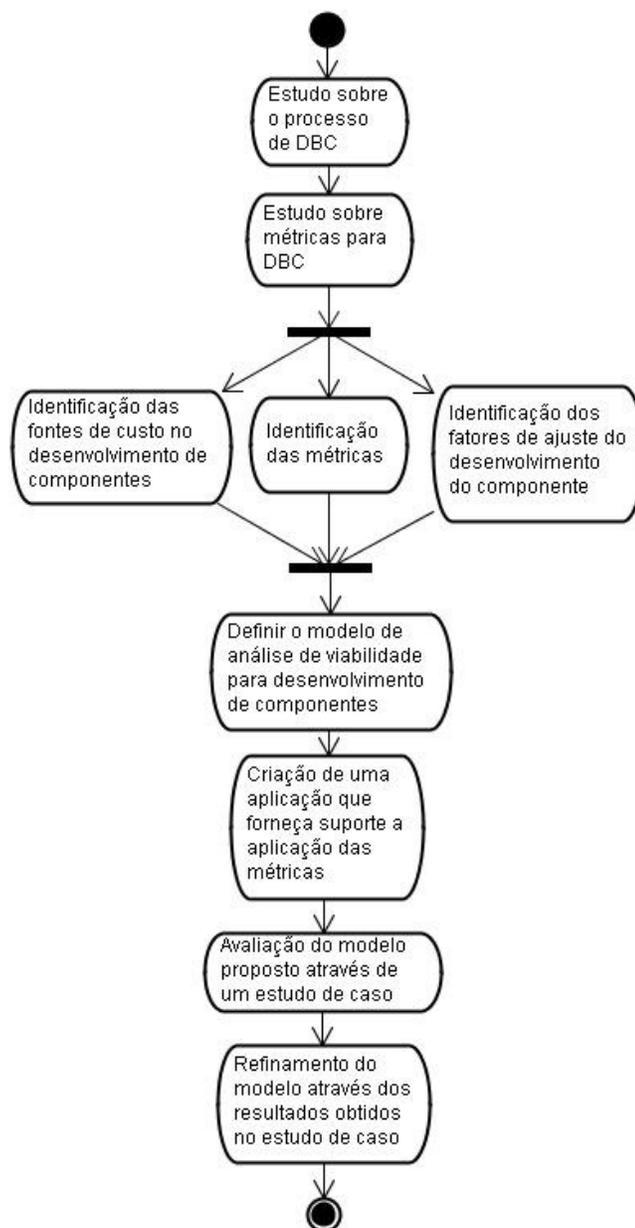


Figura 1. Desenho de Pesquisa

1.2 ESTRUTURA DO DOCUMENTO

Para a criação deste modelo, inicialmente foi conduzido um estudo acerca das atividades envolvidas no processo de desenvolvimento e reuso de componentes. Com base neste estudo (relatado na Seção 2.2), foi possível identificar as principais atividades envolvidas no processo de DBC. Essas atividades auxiliaram em um segundo estudo, relatado no Capítulo 3, que tinha por objetivo a identificação de métricas que contabilizassem o esforço envolvido na criação e no reuso de componentes. A partir das atividades e métricas estudadas foi proposto o modelo de análise de viabilidade para desenvolvimento de componentes, objeto desta dissertação.

O Capítulo apresenta uma breve análise sobre os principais métodos de análise de retorno de investimento. O Capítulo 5 apresenta propostas de modelos para análise retorno de investimento para desenvolvimento baseado em componentes. O Capítulo 6 apresenta o modelo de análise de componentização de software proposto. A análise de campo aplicada para avaliação do modelo proposto é relatada no Capítulo 7. Para finalizar, o Capítulo 8, apresenta as considerações finais sobre o presente trabalho.

2 DESENVOLVIMENTO BASEADO EM COMPONENTES

2.1 DEFINIÇÃO DE COMPONENTES

O conceito de componente forma a base para a correta compreensão do que vem a ser o desenvolvimento baseado em componentes. Embora o conceito de componente não seja novo, ainda existe muita confusão acerca do que vem a ser um componente devido ao excesso de definições existentes. Um número considerável de definições de componentes pode ser encontrado na literatura, abaixo são apresentadas algumas das principais, para que com base nelas possa-se ter uma idéia precisa do que vem a ser um componente.

Para [SZY99], “um componente de software é uma unidade de composição com interfaces contratualmente especificadas que somente podem possuir dependências explícitas de contexto. Um componente de software pode ser usado independentemente e ser composto por outras partes”.

[BAC00] define um componente como “uma implementação não conhecida de funcionalidades, que pode ser composto por outras partes em conformidade com um modelo de componentes”.

Para [CRN02], um componente é “uma unidade de composição, que deve ser especificada de tal maneira que ela possa ser composta com outros componentes e integrada dentro dos sistemas de maneira previsível”.

Analisando as similaridades das definições de [SZY99], [BAC00] e [CRN02], observa-se que componentes são uma unidade de composição com funcionalidades encapsuladas em interfaces, as quais permitem que componentes comuniquem-se com seu ambiente, podendo também serem compostos por outros componentes. Essas definições nos levam a crer que a natureza dos componentes é a de sempre serem criados para compor algo maior.

Para o presente trabalho será adotada a definição de componente de [SZY99]. Dessa forma, o modelo de análise de viabilidade para componentização de software proposto irá compreender componentes que sigam as características definidas no trabalho deste autor.

2.2 O PROCESSO DE DESENVOLVIMENTO BASEADO EM COMPONENTES

Durante a "crise do software", na Conferência de Engenharia de Software da OTAN, McIlroy defendeu a tese de que o desenvolvimento de software deveria seguir os mesmos princípios utilizados pelas outras áreas de engenharia, ou seja, o software deveria ser desenvolvido por intermédio da montagem (conexão) de componentes (rotinas) que não exigissem adaptação para necessidades específicas, e que fossem reutilizados em vários sistemas de software, adquiridos por terceiros ou desenvolvidos pela própria empresa, conforme a necessidade.

Segundo [SAM97], o desenvolvimento da maioria dos produtos de software disponíveis no mercado é baseado em uma abordagem de desenvolvimento em blocos monolíticos, formado por um grande número de partes inter-relacionadas, onde esses relacionamentos estão na maioria das vezes implícitos. O desenvolvimento baseado em componentes surgiu como uma nova perspectiva para o desenvolvimento de software, cujo objetivo é a “quebra” de blocos monolíticos em componentes interoperáveis, reduzindo desta forma a complexidade no desenvolvimento, assim como os custos, através da utilização de componentes que, a princípio, seriam adequados para serem utilizados em outras aplicações.

Para que o desenvolvimento baseado em componentes possa ser largamente utilizado, é necessário que pessoas estejam engajadas na construção de componentes genéricos que atendam as necessidades de múltiplas aplicações que serão desenvolvidas através da incorporação desses componentes. Como pode-se observar, existe a necessidade do desenvolvimento de componentes reusáveis para promover o desenvolvimento com componentes. Devido essa diferença pode-se dividir o desenvolvimento baseado em componentes em dois processos (cada um visando atender as necessidades de uma área do DBC):

- ***Desenvolvimento com componentes:*** Processo responsável por mapear as atividades necessárias para a construção de sistemas a partir de componentes reutilizáveis.
- ***Desenvolvimento de componentes:*** Processo que tem por objetivo o desenvolvimento de componentes genéricos que serão incorporados por múltiplos sistemas.

Para compreender a natureza desses processos foi conduzido um estudo na literatura visando estabelecer os conjuntos de atividades que os formam. O estudo de processos e atividades para DBC ocorreu em duas etapas. A primeira etapa do estudo focou-se na análise das atividades dos processos de desenvolvimento baseado em componentes encontrados na literatura e avalia como cada um trata as atividades consideradas elementares ao DBC. Esses processos servem como um referencial teórico para identificação das atividades base para o DBC.

Em uma segunda etapa foram estudados métodos para desenvolvimento baseado em componentes. Esses métodos foram concebidos para serem aplicados no desenvolvimento de sistemas baseados em componentes, propondo além de atividades, artefatos para a modelagem e criação do sistema. Os resultados do estudo sobre os métodos de DBC não será apresentado neste trabalho, pois nenhum dos métodos estudados comporta o desenvolvimento de componentes isoladamente do desenvolvimento de sistemas.

Nas próximas subseções serão apresentados os modelos de processos estudados. Serão abordadas somente as atividades que se focam exclusivamente no desenvolvimento de componentes, por ser o desenvolvimento de componentes o foco do presente trabalho.

2.2.1 Processos e Atividades para DBC Encontrados na Literatura

Os processos apresentados nesta subseção fornecem um importante referencial para a construção do modelo de análise de viabilidade, pois apresentam o referencial inicial sobre quais atividades devem ser consideradas no levantamento de custos inerentes a criação de componentes.

Como o presente trabalho considera que o desenvolvimento de componentes pode ocorrer independente do contexto de desenvolvimento de um sistema, ou seja, um componente pode ser identificado e construído sem estar atrelado a um projeto de sistema, foram abstraídas todas as atividades, exceto as focadas no desenvolvimento de componentes.

Abaixo segue uma breve descrição sobre os processos para desenvolvimento de componentes presentes em cada uma das propostas estudadas.

2.2.2 Abordagem de Ivica Crnkovic [CRN02]

Em [CRN02], é colocado que a Engenharia de Software apresenta grande similaridade com o desenvolvimento baseado em componentes. A grande diferença é que o DBC está especialmente focado nas questões relacionadas a componentes e desta forma distingue o desenvolvimento de componentes do desenvolvimento de sistemas com componentes. São propostas três atividades para que o processo de desenvolvimento de componentes lide com as dificuldades da geração de artefatos reutilizáveis:

- **Definição e Análise de requisitos do componente:** os problemas de analisar requisitos se tornam evidentes quando se está analisando os requisitos de um componente. Os principais fatores que tornam a análise de requisitos um processo complexo é o fato dos requisitos geralmente não estarem claros ou até mesmo não estarem completamente definidos.
- **Projeto do Componente:** essa atividade tende a ser muito complexa no desenvolvimento de componentes. O projeto de um componente deve se dar da forma mais genérica possível, primando pela adaptabilidade do componente para facilitar seu reuso. Ao mesmo tempo em que a complexidade de se projetar componentes aumenta, os componentes devem se manter simples para atenderem a requisitos específicos de uma forma eficiente. Segundo [CRN02], o desenvolvimento de um componente que visa ser reutilizado consome de três a quatro vezes mais recursos do que o desenvolvimento de um componente que serve a propósitos específicos. Para se diminuir o impacto desse risco, uma análise de mercado deve ser feita para saber se o investimento terá retorno.
- **Documentação do Componente:** como o objetivo de se criar componentes é o reuso, esses devem possuir uma documentação clara (preferencialmente geradas de uma forma padronizada). Isso é especialmente importante para componentes que sejam distribuídos na forma de código executável que podem ser mais difíceis de serem reutilizados.

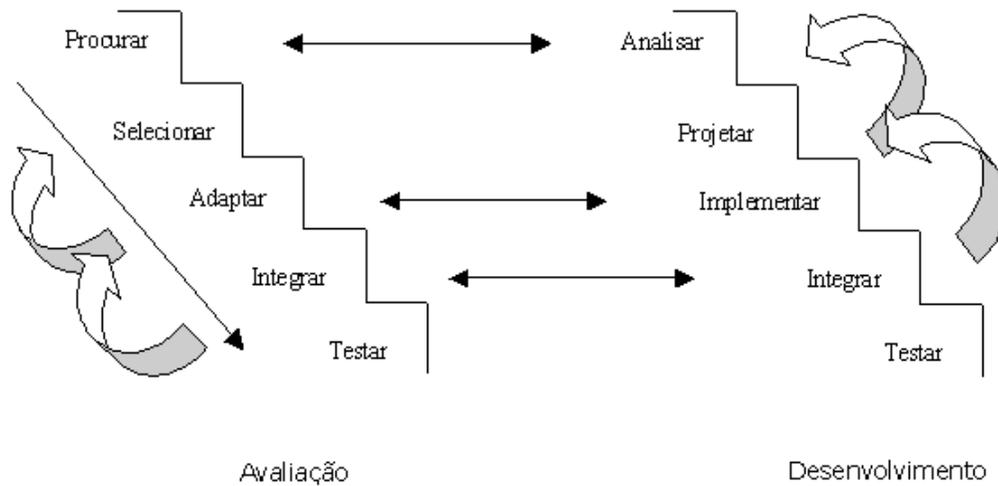


Figura 2. Processo de DBC, adaptado de [CRN02]

2.2.3 Abordagem de Johannes Sametinger [SAM97]

Segundo [SAM97], o ciclo de vida tradicional do desenvolvimento de software não suporta atividades relacionadas ao reuso de componentes. O autor cita [HOO91] e [KAN87] que definiram o refinamento do ciclo de vida do desenvolvimento de software através da inclusão das seguintes atividades:

- **Compreensão:** Compreender o problema e identificação de uma estrutura-solução baseada em componentes existentes.
- **Reconfiguração:** Reconfigurar a estrutura-solução para aumentar a possibilidade do uso de componentes existentes que estarão disponíveis na próxima fase.
- **Recuperação:** Adquirir, avaliar e instanciar componentes existentes.
- **Adaptação:** Modificar e adaptar componentes.
- **Integração:** Integrar componentes a produtos.
- **Avaliação:** Avaliar a reusabilidade de componentes criados ou gerados a partir da modificação de componentes existentes para possível incorporação ao grupo de componentes reutilizáveis existentes.

As cinco primeiras atividades estão ligadas ao desenvolvimento com reuso, e a última com o desenvolvimento para reuso. Baseado nessas atividades, [SAM97] propõe um processo de DBC baseado no modelo espiral de Boehm [BOE88].

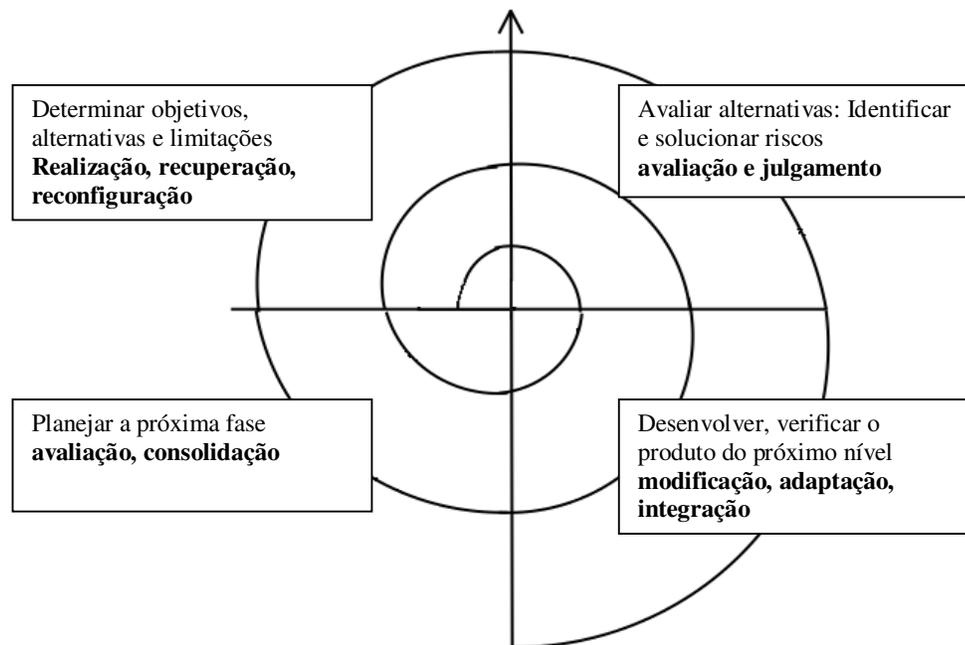


Figura 3. Modelo espiral para DBC, adaptado de [SAM97]

2.2.4 Abordagem de Klaus Bergner

O terceiro trabalho analisado foi o apresentado em [BER00], onde é proposto um processo orientado a componentes. O modelo proposto pelo autor visa dar suporte para o desenvolvimento de sistemas baseados em componentes e dos componentes que podem estar contidos neste sistema. Abaixo seguem as atividades relacionadas com o desenvolvimento de componentes:

- *Projeto de Negócios*: Essa atividade avalia os possíveis pontos de componentização de um ponto de vista de negócios.
- *Projeto Técnico*: Essa atividade avalia os possíveis pontos de componentização de um ponto de vista de técnico.

- *Especificação*: Essa atividade tem como papel principal a geração de uma completa e consistente especificação arquitetural e especificação de componentes baseada nos resultados do Projeto Técnico e do Projeto de Negócios. Como dito acima, as atividades de Projeto de Negócios e de Projeto Técnico avaliam componentes do ponto de vista técnico e de negócios, resultando numa pré-seleção de potenciais componentes para o sistema. A tarefa de Teste de Componente contém os resultados e os *logs* de testes desses componentes frente aos requisitos do sistema e a arquitetura de sistema escolhida. A tarefa Atribuição de Componentes tem como objetivo a definição de quais componentes devem ser desenvolvidos no projeto e quais componentes existentes podem ser reutilizados.
- *Implementação*: Essa atividade tem dois resultados: o código do sistema gerado pela tarefa de codificação e a validação do sistema através da tarefa de teste do sistema.

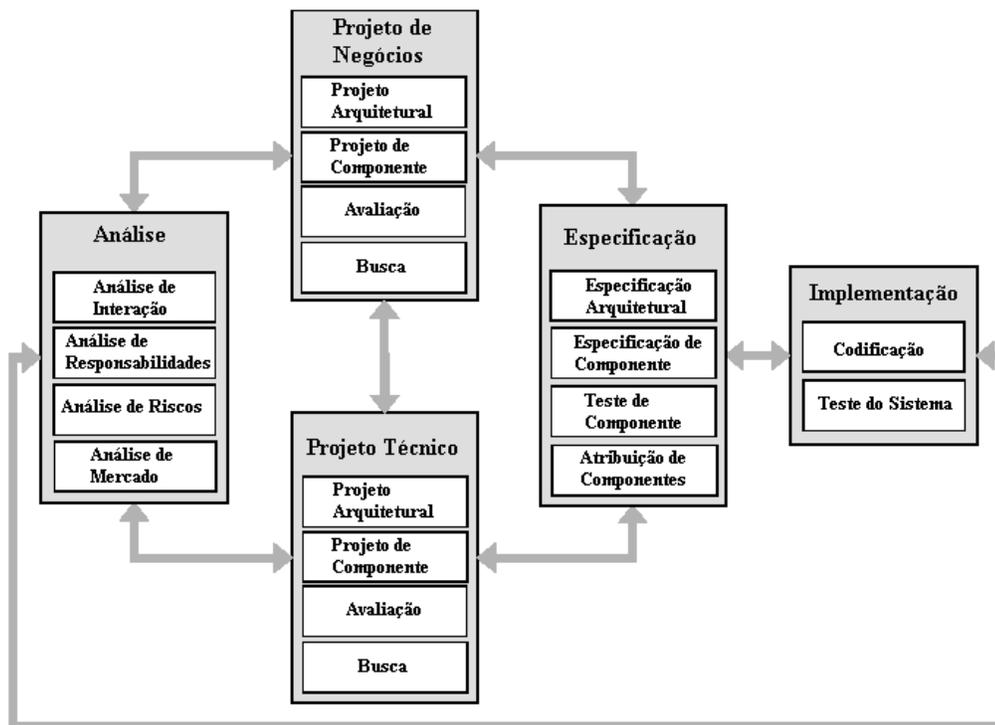


Figura 4. Modelo do processo de DBC, adaptado de [BER00]

2.3 CONSIDERAÇÕES

Através do estudo realizado sobre processos de desenvolvimento baseado em componentes pode-se analisar como a literatura aborda o processo de desenvolvimento de componentes e as suas principais atividades. Foi possível obter um conjunto de atividades, citadas por todos os autores, como as principais para o desenvolvimento baseado em componentes. Podem-se citar as atividades de Projeto e Implementação de Componente como as enfatizadas por todos os autores. Apesar de existirem atividades de mesmo nome no desenvolvimento tradicional de software, as de DBC apresentam maior complexidade e um maior número de tarefas inerentes. Esta maior complexidade pode ser justificada, pois componentes devem atender a questões de reusabilidade, generalidade e adaptabilidade.

A compreensão do processo de desenvolvimento baseado em componentes, estabelecido por esse estudo, permitiu a identificação de onde as métricas, relacionadas ao DBC (apresentadas nos próximos capítulos do presente trabalho), se encaixam dentro do contexto de Desenvolvimento Baseado em Componentes.

3 MÉTRICAS E RISCOS PARA O DESENVOLVIMENTO BASEADO EM COMPONENTES

O modelo de análise viabilidade para desenvolvimento de componentes proposto neste trabalho possui dois diferentes grupos de entradas que o alimentam. Existe um grupo de variáveis ditas quantitativas. O objetivo deste grupo de variáveis é incorporar ao modelo a parte referente à contabilização de custos referentes à componentização de software. Em complemento existe outro grupo de variáveis ditas qualitativas. As variáveis qualitativas estão relacionadas ao conceito de mapeamento de riscos no desenvolvimento de componentes. O modelo irá considerar os riscos relacionados à obsolescência do componente, ou seja, riscos inerentes ao ciclo de vida do componente que possam implicar no não reuso do artefato. No decorrer deste capítulo serão apresentadas as métricas, que refletem a parte quantitativa do modelo, e os riscos, referentes à parte qualitativa, para o desenvolvimento baseado em componentes.

3.1 MÉTRICAS PARA O DESENVOLVIMENTO BASEADO EM COMPONENTES

Reuso de componentes nem sempre implica na redução de custos de um projeto de software, da mesma forma que a criação de um componente nem sempre terá seus custos pagos. Uma série de fatores pode influenciar o retorno do investimento. No caso de componentes, a obsolescência das funcionalidades e tecnologias utilizadas em um componente podem, por exemplo, levar ao não reuso dos mesmos.

Se o uso de DBC nem sempre implica em redução de custos, a questão é definir quando DBC está ligado a custo inferior. Para responder essa questão, existem propostas de métricas. Dentre as propostas estudadas serão apresentadas aquelas cujas métricas propostas se encaixavam com o contexto do trabalho, ou seja, métricas para DBC que considerem o desenvolvimento de componentes e o reuso de componentes como processos separados.

As métricas têm um papel fundamental dentro do contexto de desenvolvimento baseado em componentes, servindo como um indicador para a tomada de decisão em um projeto. As métricas para DBC estão divididas em dois grupos [VIR03]:

- **Métricas de Processo:** Métricas de processo medem as características do processo de desenvolvimento. As medidas destas características devem revelar quando é adequada a utilização do desenvolvimento baseado em componentes (desenvolvimento com componentes e desenvolvimento de componentes).
- **Métricas de Produto:** As métricas de produto são aplicadas sobre os artefatos de um projeto de software. Essa abordagem enfoca somente sobre o produto final, não sendo considerado o seu processo de desenvolvimento. Assim sendo, essas métricas referem-se às características dos componentes. As métricas de produto podem ser largamente utilizadas, durante o processo de DBC, para se avaliar as características de componentes que se deseja integrar ao sistema.

Como o presente trabalho trabalha exclusivamente com métricas de processo, mais especificamente para o processo de decisão de quando criar componentes, não serão abordadas métricas de produto. Nas demais subseções deste capítulo serão apresentadas as métricas de processo estudadas.

3.1.1 Abordagem de Aldo Dagnino et. al. [DAG03]

Em [DAG03] é apresentado um conjunto de métricas desenvolvido e utilizado pela ABB (empresa multinacional do ramo de tecnologias de automação e produção energia). Esse conjunto de métricas tem por objetivos estabelecer quando há redução de custos e de esforço na reutilização de componentes no desenvolvimento de sistemas. Gerentes de projeto podem usar este conjunto de métricas para avaliar diferentes cenários de custos e esforço no desenvolvimento de um sistema usando a abordagem de DBC.

O conjunto de métricas foi baseado na abordagem *Goal-Question-Metrics* (Objetivos-Perguntas-Métricas). Essa abordagem estabelece um método *top-down*, no qual a partir de um grupo de objetivos se deriva um conjunto de questões que são respondidas por um conjunto de métricas cujos resultados podem determinar se os objetivos foram alcançados. Abaixo são listados os principais passos dessa abordagem:

1. Listar os principais objetivos;
2. Derivar questões necessárias para se responder se os objetivos foram alcançados. Possivelmente as questões podem ser quebradas em sub-questões até se atingir a granularidade desejada; e

3. Decidir o que deve ser medido para que as questões sejam respondidas. [DAG03] define que os dois principais objetivos para se avaliar os benefícios de DBC são:

Objetivo 1. Avaliar quando existe redução nos custos de um projeto ao se utilizar DBC; e

Objetivo 2. Avaliar quando existe redução no esforço para o desenvolvimento de um sistema baseado em componentes.

Uma vez que os objetivos foram definidos, questões e métricas podem ser estabelecidas. Para definir as questões e as métricas foi utilizado o conceito de unidades de medida (UM). Para [DAG03], o tamanho de um projeto de software é a entrada chave para a estimativa de custo e esforço, este tamanho é representado em UM. O autor cita duas medidas convencionais para se estimar o tamanho de um software:

- Linhas de código (LOC): LOC é a soma das linhas não comentadas (NLOC) e das linhas comentadas de código (CLOC); e
- Pontos de função (FP): A abordagem de FP divide o software em cinco conjuntos de funcionalidade que possuem pesos diferentes: entradas, saídas, arquivos de dados, interface e consultas. Para se realizar a estimativa, o número de tipos de cada funcionalidade é somado. Esse número multiplicado pelo peso da funcionalidade produz o valor do ponto de função.

A seguir serão apresentadas as questões e métricas para cada objetivo. Ao lado de cada pergunta estará abreviada a unidade de resposta para a mesma.

Para medir o custo de forma apropriada é necessário saber o esforço envolvido no desenvolvimento do software, as métricas dos dois objetivos acabam se misturando para trazer maior precisão aos valores medidos. Na Figura 5 são apresentados os fluxos que levam os objetivos até as métricas. Os nodos ovais representam os parâmetros específicos ao projeto, e os no formato de um paralelogramo representam os parâmetros de entrada que são específicos da organização.

- Objetivo 1: Avaliar quando existe redução nos custos de um projeto ao se utilizar DBC.
 - Questão 1: Qual a diferença de custo entre o DBC e o desenvolvimento tradicional do software?
 - Questão 1.1: Qual o custo de mão de obra por hora para produzir uma UM de código (\$/UM)?
 - Métrica 1: Determinar a diferença do custo de mão de obra por hora entre as diferentes organizações (que desenvolvem e reusam o componente) (\$/hora).
 - Métrica 2: Determinar a taxa de produtividade de software da organização alvo (UM/hora).

O custo de mão de obra por hora e a taxa de produtividade variam entre as organizações. O custo de mão de obra por unidade de medida pode ser calculado se dividindo a métrica 1 pela métrica 2. O custo de mão de obra por unidade de medida depende dos custos por hora e da taxa de produtividade por hora.

- Questão 1.2 (ou objetivo 2): Qual é a diferença de esforço entre o DBC e o desenvolvimento tradicional na criação de um software?
- Questão 1.2.1: Qual é o esforço requerido no DBC? (UM)
- Questão 1.2.1.1: Qual é o esforço requerido para se desenvolver uma fração da aplicação se usando a abordagem tradicional?
 - Métrica 3: Tamanho do esforço requerido para se desenvolver uma fração de código de uma forma tradicional (tal como *glue code*¹) requerido pela abordagem de DBC. Esse pedaço de código em unidades de medida é o mesmo valor do tamanho da fração de código em unidades de medida. Por exemplo, se o novo código tem um tamanho de 100 UM, então o esforço estimado é de 100 UM.

¹ Fragmento de código criado para conectar componentes.

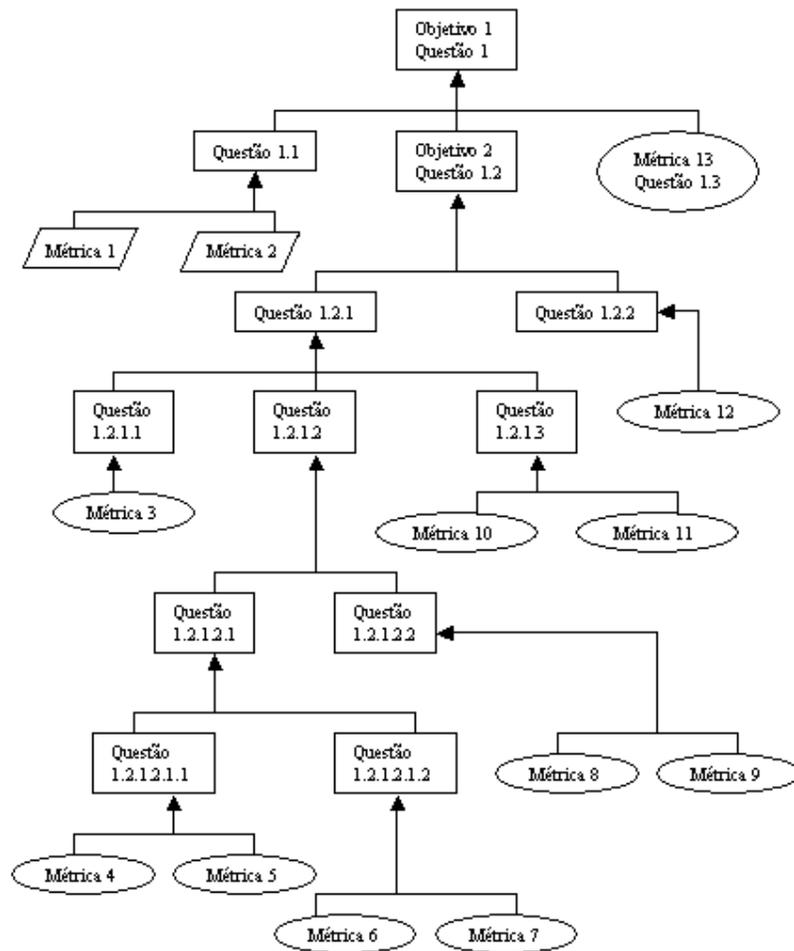


Figura 5. Abordagem GQM para métricas de DBC, adaptado de [DAG03]

- Questão 1.2.1.2: Qual o esforço requerido (ER) para se reutilizar componentes? (UM)

O esforço para se reutilizar componentes é a soma dos esforços de se reutilizar componentes que necessitam de modificações mais o esforço de se reutilizar componentes que não necessitam modificações.

- Questão 1.2.1.2.1: Qual o esforço requerido para se reutilizar componentes com modificações? (UM)

Deve ser determinado o esforço para se reutilizar componentes oriundos de um repositório de componentes ou criados por terceiros. O esforço em se reutilizar componentes com modificações varia conforme o grau das modificações. Em [DAG03], o grau de modificações é dividido em dois níveis, um com modificações menores ou iguais a 25% das funcionalidades do componente e outro de modificações superiores a 25%. Por exemplo, se há um componente de 600 UM e 140 UM devem ser modificadas antes do seu reuso, o nível de modificação é inferior a 25%.

- Questão 1.2.1.2.1.1: Qual é o esforço requerido para se reutilizar componentes que requerem modificações menores ou iguais a 25% das funcionalidades do componente? (UM)
 - Métrica 4: O tamanho acumulado de todos os componentes reutilizados com modificações menores ou iguais a 25% das funcionalidades do componente. (UM)
 - Métrica 5: Definir o fator ERLM de todos os componentes reutilizados com modificações menores ou iguais a 25% das funcionalidades do componente. (%)

O fator ERLM representa o fator de esforço de reuso de componentes com modificações menores ou iguais a 25% (LM vem do inglês, *less modification*). Esse fator inclui o esforço requerido para se selecionar, compreender, integrar, testar e modificar o componente. [DAG03] cita [RAJ94] para estimar que o esforço requerido para se reutilizar um componente que requer modificações menores ou iguais a 25% é de 40% do esforço requerido para se escrever o mesmo componente utilizando uma abordagem tradicional. Esse valor serve somente como referência, os gerentes de projeto podem modificá-lo baseado em projetos passados da organização. A Equação 1 apresenta a função para o cálculo do esforço de reutilizar componentes com modificações menores ou iguais a 25% das funcionalidades do componente.

$$ER_{(m \leq 25\%)} = TC \times ERLM \quad (1)$$

Onde:

$ER_{(m \leq 25\%)}$: esforço total para se reutilizar um componente com modificações menores ou iguais a 25%.
 TC: soma dos tamanhos dos componentes que sofreram alterações inferiores ou iguais a 25% das funcionalidades do componente.

ERLM: fator de esforço de reuso de componentes com modificações menores ou iguais a 25%.

- Questão 1.2.1.2.1.2: Qual o esforço requerido para se reutilizar componente com modificações superiores a 25% das funcionalidades do componente?
(UM)
 - Métrica 6: O tamanho acumulado de todos os componentes reutilizados com modificações superiores a 25% das funcionalidades do componente.
(UM)
 - Métrica 7: Definir o fator ERMM de todos os componentes reutilizados com modificações superiores a 25% das funcionalidades do componente.
(%)

O fator ERMM representa o fator de esforço de reuso de componentes com modificações superiores a 25% das funcionalidades do componente (MM vem do inglês, *more modification*). Assim como o ERLM, o ERMM representa o esforço requerido para se selecionar, compreender, integrar, testar e modificar o componente. Novamente [DAG03] cita [RAJ94] para estimar que o esforço requerido para se reutilizar um componente que requer modificações superiores a 25% é de 90% do esforço requerido para se escrever o mesmo componente utilizando uma abordagem tradicional. A Equação 2 apresenta a função para o cálculo do esforço de reutilizar componentes com modificações superiores à 25% das funcionalidades do componente.

$$ER_{(m>25\%)} = TC \times ERMM \quad (2)$$

Onde:

$ER_{(m>25\%)}$: esforço total para se reutilizar um componente com modificações superiores a 25%.

TC: soma dos tamanhos dos componentes que sofreram alterações superiores a 25% das funcionalidades do componente.

ERMM: fator de esforço de reuso de componentes com modificações superiores a 25%.

Para se calcular o esforço total de reutilização de componentes com modificações é realizada a soma dos esforços de reutilização de componentes com modificações superiores e inferiores a vinte e cinco por cento, apresentada na Equação 3.

$$ETRM = ER_{(m \leq 25\%)} + ER_{(m > 25\%)} \quad (3)$$

Onde:

ETRM: esforço total para se reutilizar componentes com modificações.

$ER_{(m \leq 25\%)}$: esforço total para se reutilizar um componente com modificações menores ou iguais a 25%.

$ER_{(m > 25\%)}$: esforço total para se reutilizar um componente com modificações superiores a 25%.

- Questão 1.2.1.2.2: Qual o esforço requerido para se reutilizar componentes sem modificação? (UM)
 - Métrica 8: O tamanho acumulado de todos os componentes reutilizados sem modificações. (UM)
 - Métrica 9: Definir o fator ERNM de todos os componentes reutilizados sem modificações. (%)

O fator ERNM representa o fator de esforço de reuso de componentes sem modificações (NM vem do inglês, *not require modification*). Assim como o ERLM, o ERMM representa o esforço requerido para se selecionar, compreender, integrar e testar e o componente. [DAG03] cita os estudos de caso de [GRA92], [LIM94] e [MAR92] para estimar que o esforço requerido para se reutilizar um componente que não requer modificações é de 20% do esforço requerido para se escrever o mesmo componente utilizando uma abordagem tradicional.

A Equação 4 apresenta a função para o cálculo do esforço de reutilizar componentes que não necessitam sofrer modificações.

$$ETRSM = TC \times ERNM \quad (4)$$

Onde:

ETRSM: esforço total para se reutilizar componentes sem modificações.

TC: tamanho acumulado de todos os componentes reutilizados sem modificações.

ERNM: fator de esforço de reuso de componentes sem modificações.

Baseado nas métricas anteriormente apresentadas é estabelecido que o esforço total de reuso de componentes é a soma do esforço total para se reutilizar componentes sem modificações com o esforço total para se reutilizar componentes com modificações, apresentado na Equação 5.

$$ETR = ETRM + ETRSM \quad (5)$$

Onde:

ETR: esforço total de reuso de todos os componentes.

ETRM: esforço total para se reutilizar componentes com modificações.

ETRSM: esforço total para se reutilizar componentes sem modificações.

- Questão 1.2.1.3: Qual é o esforço requerido para se escrever componentes reutilizáveis? (UM)
 - Métrica 10: Tamanho do componente escrito para ser reutilizado (UM). Esta métrica corresponde ao tamanho das funcionalidades do novo componente.
 - Métrica 11: O fator ERE (%).

Fator ERE: O esforço requerido para se escrever um componente reutilizável é consideravelmente maior do que o de se desenvolver exatamente as mesmas funcionalidades em uma abordagem tradicional. O esforço adicional vem da necessidade de se conduzir uma análise de domínio, suportar requisitos de forma mais genérica, testes adicionais, empacotamento, manutenção, certificação, documentação e gerenciamento do repositório. Todos estes fatores estão incorporados no fator ERE. Novamente [DAG03] cita diversos estudos sobre o desenvolvimento de componentes e sugere um fator de 1.55, ou seja, o esforço para se desenvolver um componente reutilizável é 55% a mais do que o de se desenvolver tradicionalmente. A função de cálculo do esforço para desenvolvimento de componentes reutilizáveis é apresentada na Equação 6.

$$ETER = TC \times ERE \quad (6)$$

Onde:

ETER: esforço total para se escrever um componente reutilizável.

TC: tamanho das funcionalidades a serem componentizadas.

ERE: esforço extra requerido para se escrever um componente reutilizável

Partindo das métricas anteriormente apresentadas, [DAG03] estabelece que o esforço total para desenvolver um projeto usando DBC é representado pela soma do esforço total para se desenvolver frações de código usando a abordagem tradicional, esforço total de se reutilizar componentes e esforço total para escrever componentes reutilizáveis, como apresentado na Equação 7.

$$ETDBC = TECN + ETR + ETER \quad (7)$$

Onde:

ETDBC: esforço total para desenvolver um projeto usando DBC.

TECN: esforço total para se desenvolver frações de código usando a abordagem tradicional.

ETR: esforço total de se reutilizar componentes

ETER: esforço total para escrever componentes reutilizáveis.

- Questão 1.2.2: Qual é o esforço necessário para se desenvolver o sistema utilizando a abordagem tradicional de desenvolvimento de software?
 - Métrica 12: O esforço necessário para se desenvolver o módulo ou produto em questão, se utilizando a abordagem tradicional, é igual ao tamanho do mesmo, como apresentado na Equação 8. Por exemplo, se o tamanho de um produto é 1000 UM, então o total de esforço para se desenvolver um projeto utilizando a abordagem tradicional de desenvolvimento é de 1000 UM.

$$ETT = TP \quad (8)$$

onde:

ETT: esforço necessário para se desenvolver o sistema utilizando a abordagem tradicional.

TP: tamanho do software em alguma unidade de medida de software.

A diferença de esforços entre a abordagem tradicional e a baseada em componentes pode ser obtida se calculando o ET_{salvo} , apresentado na Equação 9.

$$ET_{salvo} = ETT - ETDBC \quad (9)$$

onde:

ET_{salvo} : esforço poupado pelo reuso de componentes.

ETT: esforço necessário para se desenvolver o sistema utilizando a abordagem

ETDBC: esforço total para desenvolver um projeto usando DBC.

A variável ET_{salvo} será o indicador de benefício ou perna de esforço no uso do desenvolvimento baseado em componentes no desenvolvimento do sistema. Caso o ET_{salvo} seja maior que zero, então o uso de DBC irá requerer menor esforço do corpo técnico da empresa. Caso ET_{salvo} seja menor que zero, então a abordagem de desenvolvimento do sistema é a tradicional irá requerer menor esforço.

Do conjunto de métricas proposto em [DAG03], as utilizadas no modelo de análise de viabilidade são:

- Custo de Desenvolvimento do Componente;
- Custo de Reuso do Componente, e
- Custo de Não Reutilizar o Componente - Custo de Desenvolvimento.

3.1.2 Abordagem de Jeffrey Poulin e Joseph Caruso [POU93]

Em [POU93], é apresentado um conjunto de métricas utilizadas pela IBM para se estimar o esforço salvo através do reuso. As métricas apresentadas neste trabalho seguem a perspectiva de negócios, sempre lidando com custos e benefícios referentes ao reuso. Apesar de [POU93] apresentar um conjunto extenso de métricas, serão apresentadas somente as métricas relacionadas ao escopo de métricas de processo.

Em [POU93], é proposta uma métrica que indica quando é adequado reutilizar componentes e outra de quando é adequado se criar componentes reutilizáveis. Essas métricas são apresentadas nas seções abaixo.

3.1.2.1 *Custo poupado pelo reuso*

O custo poupado pelo reuso (CPPR) é a métrica responsável por calcular o decréscimo do custo do produto em virtude do reuso. Essa métrica possui um papel fundamental por ser ela o indicador do retorno do investimento em reuso.

Os benefícios potenciais e os custos que podem ser poupados através do reuso variam conforme o projeto e o seu percentual de reuso. Mesmo organizações que somente consomem componentes devem realizar investimentos em mudanças de processos, ferramentas e treinamentos. Por exemplo, desenvolvedores e gerentes irão necessitar receber treinamento sobre reuso de software, esse treinamento irá despende uma certa quantidade de recursos. Sem esse treinamento os consumidores de componentes não iriam conseguir usufruir todos os benefícios oferecidos pelo reuso. Para se estimar o CPPR é proposto o seguinte somatório:

$$\text{CPPR} = \sum_{i=1}^j b_i = \sum_{i=1}^k c_i \quad (10)$$

Onde:

b_i : é o benefício de ser o consumidor deste componente.

c_i : é o custo associado a se ser o consumidor deste componente.

Os custos e benefícios de se adotar o reuso variam conforme o projeto e a organização. Os benefícios de se reutilizar software geralmente superam os seus custos, pois são evitados gastos com projeto de parte do software, desenvolvimento, documentação, teste, manutenção e gerenciamento do desenvolvimento do artefato reutilizável. Benefícios também podem incluir a redução de custos na aquisição de ferramentas ou equipamentos que seriam necessários caso o software tivesse de ser inteiramente desenvolvido na organização. Outros benefícios que vem agregados com a adoção de DBC são o aumento de lucros pela diminuição de tempo de produção do produto e/ou aumento da satisfação dos clientes.

Os custos referentes ao reuso incluem gastos como o dinheiro gasto para educar a organização a reutilizar software e na necessidade de se adquirir ferramentas que disponibilizem informações sobre os componentes reutilizáveis da empresa. A organização pode também incorrer custos pela análise de sistema e de domínio e tempo gasto na identificação de partes do sistema que podem ser componentizadas. Outro gasto necessário é o tempo gasto pelos desenvolvedores para se selecionar os componentes candidatos a reuso. Se a organização tiver adquirido componentes, pode haver gastos com o pagamento de taxas de licença ou a compra do componente.

[POU93] estima que o benefício financeiro atribuído ao reuso durante a fase de desenvolvimento é de 80% do custo de se desenvolver o código integralmente. Essa porcentagem deriva de estudos que demonstraram que o custo de se integrar um componente é em torno de 20% do custo de se desenvolver um módulo de software novo. Esse mesmo estudo também demonstra que os gastos na fase de manutenção do sistema serão menores, pois componentes estão sujeitos a possuírem menos erros. O custo poupado pelo reuso (apresentado na Equação 13) é calculado pela soma dos custos evitados nas fases de desenvolvimento (apresentado na Equação 11) e de manutenção (apresentado na Equação 12).

$$CPD = CR \times 0.8 \times CCN \quad (11)$$

Onde:

CPD: Custo poupado no desenvolvimento.

CR: código reutilizado.

CCN: custo do código novo.

$$CPM = CR \times TE \times CMC \quad (12)$$

Onde:

CPM: Custo poupado na manutenção.

CR: código reutilizado.

TE: taxa de erros por unidade de medida de software.

CMC: custo de manutenção do código.

$$CPPR = CPD + CPM \quad (13)$$

Onde:

CPPR: Custo poupado pelo reuso.

CPD: Custo poupado no desenvolvimento.

CPM: Custo poupado na manutenção.

Algumas das variáveis apresentadas nas fórmulas acima variam conforme a organização, e devem ser obtidos a partir de seus dados históricos. A seguir é apresentado um exemplo que ilustra a aplicação das métricas de avaliação do custo poupado pelo reuso. Como exemplo da aplicação dessas métricas, digamos que uma organização tem o custo de desenvolvimento histórico de \$4 por linha de código, taxa de erro de 1.25 erros a cada mil linhas de código e \$100 de média para arrumar um erro. O custo poupado pelo reuso para se integrar um componente de 20 KLOC nesse cenário seria de:

$$\begin{aligned} CPPR &= (20 \text{ KLOC} \times 0.8 \times \$4 \text{ por linha}) \\ &+ (20 \text{ KLOC} \times 1.25 \text{ erros por KLOC} \times \$100 \text{ por erro}) \\ &= 64 \text{ mil} + 2.5 \text{ mil} \\ &= 66.5 \text{ mil} \end{aligned}$$

3.1.2.2 Custo de desenvolver componente

Os custos relacionados a se desenvolver um componente reutilizável claramente são superiores ao de se desenvolver de uma forma tradicional. Muito esforço deve ser despendido para que um componente se torne facilmente reutilizável em diversos domínios de aplicação. A métrica de custo adicional de desenvolvimento (CAD) visa quantificar esse esforço, como apresentado na Equação 14.

Para determinar o CAD deve-se inicialmente identificar e quantificar os custos e os benefícios associados com a produção de componentes reutilizáveis.

$$CAD = \sum_{i=1}^j c_i - \sum_{i=1}^k b_i \quad (14)$$

Onde:

b_i : é o benefício de ser o produtor de componentes.

c_i : é o custo associado a se ser o produtor de componentes.

O grau do investimento na construção de componentes reutilizáveis varia conforme as prioridades e necessidades da organização. Abaixo segue uma lista com os principais custos relacionados ao desenvolvimento de componentes reutilizáveis.

- Análise do domínio é requerida para se conduzir um estudo completo do problema que pode revelar oportunidades de artefatos a serem componentizados.
- Treinamento requerido para que os desenvolvedores de software aprendam os conceitos e as práticas de se construir software reutilizável. O treinamento em reuso de software deve incluir conceitos como encapsulamento, ocultamento de atributos, construção de interfaces bem definidas, o uso de funcionalidades específicas de cada linguagem e programação com independência de ambiente.
- Necessidade de se possuir um repositório de componentes que armazene os diversos componentes reutilizáveis produzidos pela empresa. Também podem ser contabilizados gastos com a manutenção desse repositório.
- Desenvolvimento e certificação dos componentes reutilizáveis para assegurar-se de que os artefatos tenham sido projetados, executados, e testados para serem reutilizados em outros ambientes. A organização deve também dar ênfase extra na documentação do usuário para que o reutilizador do componente possa compreender suas funcionalidades e interfaces. Para os componentes de alta qualidade, reusáveis, nós requeremos a certificação por um grupo independente do teste assegurar a qualidade e a função do software.

Esses são somente alguns dos principais custos relacionados ao desenvolvimento de componentes reutilizáveis. Felizmente, muitos desses custos representam um investimento inicial que deve ser feito, não sendo necessário refazê-los novamente.

Assim como na definição do CPPR, os autores acharam difícil quantificar todos os benefícios e custos relacionados ao reuso, portanto, foi estabelecido um valor padrão, baseado em experiências passadas, para definir o esforço relativo a se escrever componentes reutilizáveis. Esse valor estabelece que o esforço para se desenvolver um componente reutilizável é de 150% do esforço de se escrever as mesmas funcionalidades sem focar no reuso. Dessa forma, a fórmula para se calcular o custo adicional de desenvolvimento de um componente reutilizável é apresentada na Equação 15.

$$CAD = (CRER - 1) \times QCC \times CDCN \quad (15)$$

Onde:

CAD: custo adicional de desenvolvimento de um componente reutilizável.

CRER: custo relativo de se escrever para o reuso.

QCC: quantidade de código do componente.

CDCN: custo de se desenvolver código novo.

Para se ilustrar a aplicação da métrica apresentada acima considere o seguinte exemplo. Digamos que uma empresa vai desenvolver um componente que terá em torno de 10 KLOC, tem o custo de desenvolvimento histórico de \$4 por linha de código e o custo relativo a se escrever para o reuso é de 1.5. Dessa forma, o custo adicional de se escrever esse componente para ser reutilizado é:

$$\begin{aligned} CAD &= (1.5 - 1) \times 10 \text{ KLOC} \times \$4 \text{ por linha de código} \\ &= \$20 \text{ mil} \end{aligned}$$

Do conjunto de métricas proposto em [POU93], as utilizadas no modelo de análise de viabilidade são:

- Custo de Desenvolvimento do Componente;
- Custo de Não Reutilizar o Componente - Custo de Manutenção.

3.1.3 Considerações

O Capítulo 3 abordou de forma sucinta duas propostas de grupos de métricas para análise de reuso de componentes. Essas duas propostas servem de base para o modelo de análise de viabilidade proposto.

Através da composição de diversas das métricas citadas, em adição a fatores qualitativos, o modelo de análise de viabilidade foi criado. As métricas apresentadas nesta seção e que foram utilizadas para criação do modelo estão apresentadas na Tabela 1.

Tabela 1. Métricas utilizadas na composição do modelo

Métrica para cálculo de:	Baseada na proposta de:	Número da Equação:
Custo de Desenvolvimento do Componente	[DAG03]	2, 3 e 4
	[POU93]	15
Custo de Reuso do Componente	[DAG03]	5
Custo de Não Reutilizar o Componente - Custo de Desenvolvimento	[DAG03]	8
Custo de Não Reutilizar o Componente - Custo de Manutenção	[POU93]	12

Durante o capítulo 6, no qual o modelo proposto será explicado, serão referenciadas as métricas apresentadas na presente seção.

3.2 RISCOS PARA O DESENVOLVIMENTO BASEADO EM COMPONENTES

Segundo [CRN02], componentes seguem um ciclo de vida que vai da sua concepção, reutilização até o momento que as funcionalidades providas pelo artefato começam a ficar desatualizadas até se tornarem obsoletos, conseqüentemente são descontinuados e não mais reutilizados. Esta seção visa estabelecer algumas das características que podem influenciar sobre o tempo de vida do componente, ou seja, até quando um componente poderá suprir as necessidades das aplicações que o reusam. O mapeamento destes riscos permite ao engenheiro de software validar quando os riscos inerentes a obsolescência do componente podem reduzir a sua probabilidade de se manter reutilizável. Considerar os riscos torna-se especialmente importante nos casos em que o custo do desenvolvimento de um componente é amortizado em projetos realizados com grandes janelas de tempo, por exemplo, projetos com diferença de início de mais de doze meses.

Para a definição de quais riscos seriam incorporados ao modelo foi conduzida uma pesquisa na literatura acerca de riscos no desenvolvimento de componentes. Abaixo são apresentadas breves descrições sobre os trabalhos estudados.

- [FRA96a] classificou a falha no reuso de software (que pode-se considerar a manifestação de um risco) em sete grupos de falhas distintas e, como a parte de uma pesquisa conduzida, que o grupo de falha a mais importante, ou a mais presente, era que ninguém implementava para reuso. Em 2001, entretanto, [FIC01] reagrupou os grupos de falhas em três categorias mais genéricas: candidato a reuso não procurados (grupo 1 de [FRA96a]); o candidato a reuso não pode ser reutilizado (grupos 2 a 4 de [FRA96a]); e candidato a reuso não encontrado (grupos de falha 5 a 7 de [FRA96a]). Baseado nestes três agrupamentos, uma conclusão diferente pode ser alcançada, isto é, os grupos de riscos mais importantes ou mais presentes são: que (1) os candidatos a reuso não podem ser usados e (2) os candidatos reusar não podem ser encontrados. Como o modelo considera somente os riscos relevantes à obsolescência do componente, serão considerados os riscos do grupo “os candidatos a reuso não podem ser reutilizados”.
- O Software Productivity Consortium (SPC) desenvolveu e publicou uma estratégia de adoção de reuso [SPC93], como uma ferramenta para auxílio de implantação de processos de reuso de software. Neste trabalho é apresentado um modelo de processo para criação de artefatos reutilizáveis destacando os riscos a serem considerados para cada atividade.
- Em [EZR02] é apresentado um número de pontos que devem ser considerados antes da tomada de decisão a respeito da implantação de um processo de reuso de software. Estes pontos devem ser avaliados no contexto de atividades de gerência de risco. Para [EZR02], antes de criar um componente reutilizável, necessita-se compreender que o desenvolvimento de um artefato reutilizável representa um investimento. Conseqüentemente, deve ser conduzido um processo de tomada de decisão para minimizar os riscos. Para tal, é proposto um grupo de fatores a serem considerados no processo de tomada de decisão.
- Em [LIM98] é apresentado um extenso trabalho sobre reuso de software. Assim como os demais trabalhos citados, [LIM98] destaca uma série de riscos ligados ao desenvolvimento de artefatos reutilizáveis.

Do grupo de riscos, propostos pelos autores, foram extraídos fatores ligados ao ciclo de vida do componente, e que conseqüentemente podem levar a não reutilização. A seguir são

apresentados os grupos de riscos ligados à obsolescência dos componentes levantados.

- **Manutenção:** Risco relacionado à falta de manutenção no componente após a sua criação. Desta forma, dificultando a correção de *bugs* e evolução do mesmo.
- **Documentação:** Como componentes são artefatos caixa-preta, a documentação sobre o componente e suas interfaces é extremamente importante para facilitar o reuso do mesmo. Mas existe o risco da documentação estar incompleta ou possivelmente não existir, desta forma dificultando o processo de reuso.
- **Performance:** Componentes, assim como hardware, tendem a se tornarem menos performáticos com o passar do tempo. Este risco está diretamente ligado à possibilidade do componente não ser reutilizado, pois a sua performance não atende mais as necessidades das aplicações que poderiam o reutilizar.
- **Obsolescência Funcional:** O risco da obsolescência funcional está diretamente ligado aos serviços providos por um componente. Por exemplo, se compararmos um componente para validação de número de cartão de crédito a um componente para cálculo de imposto de renda. O componente de cálculo de imposto tende a se tornar obsoleto mais rapidamente, pois cobranças de impostos mudam mais frequentemente do que um algoritmo de validação de cartão de crédito.
- **Obsolescência Técnica:** A obsolescência técnica visa abranger os riscos relacionados às tecnologias necessárias para reutilizar o componente, ou seja, tecnologias que de alguma forma provêm suporte ou ambiente para reuso e execução componente. Dentre essas tecnologias podemos citar sistema operacional, linguagem de programação, banco de dados, dentre outras.

3.2.1 Considerações

De posse de um grupo fechado de riscos relacionados à obsolescência do componente, os mesmos foram validados frente às propostas estudadas. A Tabela 2 apresenta o relacionamento dos riscos propostos por cada autor.

Tabela 2. Fatores que influenciam o reuso de componentes

	[FIC01]	[SPC93]	[LIM98]	[EZR02]
Manutenção				
Documentação				
Performance				
Obsolescência Funcional				
Obsolescência Técnica				

Com base nesse grupo de riscos, o modelo proposto poderá integrar fatores qualitativos as suas entradas. Dessa forma, podendo gerar resultados refinados quanto à análise de retorno de investimento.

4 ANÁLISE FINANCEIRA PARA CRIAÇÃO DE COMPONENTES

Existe um número considerável de métodos que são utilizados para avaliar projetos de investimento, variando desde a pura intuição do empresário aos mais sofisticados modelos matemáticos. Todavia, nem todas essas técnicas são fundamentadas em bases conceitualmente corretas, e, naturalmente, as orientações decorrentes da abordagem de um dado problema através da aplicação de algumas delas podem confundir o responsável pela decisão, induzindo-o ao erro [OLI82].

A análise de investimento em reuso, no domínio da análise financeira corporativa, é uma área cujas preocupações não são atendidas diretamente pela engenharia de software. Um ponto importante a ser levantado é que, no contexto de análise financeira corporativa, o investimento em reuso é somente uma alternativa para a empresa. Em um contexto corporativo, a análise de investimento somente se refere à melhor forma de alocar os capitais e recursos humanos da empresa [FAV96].

Partindo desse contexto, sempre existe uma alternativa à implantação de um programa de reuso, como por exemplo, investir capital equivalente no mercado financeiro, que provê uma taxa de retorno anual esperada. Como investimentos no mercado financeiro são analisados em relação a períodos de tempo (e.g. o retorno esperado de um fundo de investimento em sete meses), investimentos em reuso devem ser analisados da mesma forma, para poder ser estabelecido um comparativo entre ambos.

O presente capítulo apresenta algumas técnicas de análise financeira. Essas técnicas servirão como base para a criação do modelo de análise de viabilidade para desenvolvimento de componentes.

4.1 PAYBACK

O método Payback determina o tempo necessário para recuperar os recursos investidos em um projeto, como por exemplo, a criação de um componente reutilizável. Quanto mais amplo for o horizonte de tempo considerado, maior será o grau de incerteza nas previsões. Desse modo, propostas de investimento com menor prazo de retorno apresentam maior liquidez e, conseqüentemente, menor risco.

O cálculo do *Payback* é realizado da seguinte forma:

- Se as entradas líquidas de caixa forem uniformes, bastará dividir o investimento inicial pelas entradas anuais de caixa;
- Quando as entradas anuais forem desiguais, estas deverão ser acumuladas até atingir o valor do investimento, apurando-se o prazo de retorno.

No contexto de desenvolvimento de componentes *Payback* pode ser visto como o número de reusos necessários para se recuperar investimento na criação do componente. Em [BAR87] e [SPC92], o *Payback* na criação de componentes é definido por:

$$N_0 = \frac{E}{1 - B} \quad (16)$$

Onde:

N_0 : Número de vezes que um componente deve ser reutilizado para ter seu custo pago;

E: Custo de se desenvolver um componente reutilizável; e

B: Custo de se integrar um componente.

A idéia de *Payback*, ou recuperação de custos é naturalmente muito apelativa, mas para análises de investimento essa abordagem apresenta diversos problemas, como por exemplo [FAV96]:

- A definição da data de retorno de investimento é em geral arbitrária e subjetiva;
- Não são reconhecidas as entradas de caixa previstas para ocorrerem após a recuperação do investimento; e
- O método não avalia adequadamente o valor do dinheiro no tempo.

O método *Payback* é valioso para comunicação da noção de valor agregado de um projeto. Sua facilidade e natureza intuitiva facilitam a propagação dos benefícios do reuso para gerentes não acostumados as estimativas de custo. Ou seja, *Payback* é indicado para demonstração dos resultados de investimentos feitos, mas não é adequado para a própria análise de investimento.

4.2 VALOR PRESENTE LÍQUIDO

O método do Valor presente Líquido (VPL), também conhecido pela terminologia Método do Valor Atual Líquido, caracteriza-se, essencialmente, pela transferência para o instante presente de todas as variações de caixa esperadas, descontada a taxa mínima de atratividade (apresentada na Seção 4.2.1). Em outras palavras, seria o transporte para a data zero de um diagrama de fluxos de caixa, de todos os recebimentos e desembolsos esperados, descontada a taxa de juros considerada. Abaixo segue a equação para o cálculo do VPL.

$$VPL = \sum_{t=1}^n \frac{FC_t}{(1+k)^t} - II \quad (17)$$

Onde:

FC_t – valor presente das entradas de caixa;

t – período de ocorrência de fluxo de caixa;

n – total de períodos;

k – taxa mínima de atratividade; e

II – Investimento Inicial.

Um valor presente líquido positivo significa que o valor de todos os fluxos de caixa positivos é o maior que o correspondente às variações negativas de caixa. Dessa forma, um projeto cujo valor presente líquido seja positivo, será economicamente interessante para a taxa de atratividade considerada, tornando-se tanto mais atrativo quanto maior for o seu valor presente líquido. Essa afirmação leva a conclusão de que quando se compara alternativas, aquela de maior valor presente líquido deve ser a preferida sob o ponto de vista econômico.

Um valor presente líquido nulo significa que o projeto, embora a taxa de desconto considerada, não é desinteressante de um ponto de vista econômico; produz um retorno de valor apenas igual ao custo do capital, deixando de ser compensadora a sua implementação.

Valor presente líquido negativo indica que o projeto deve ser rejeitado, pelo fato de que os benefícios financeiros prospectivos não serem suficientes para assegurar sequer a recuperação do investimento realizado.

4.2.1 Taxa Mínima de Atratividade

A principal característica do método de Valor Presente Líquido é o reconhecimento da variação do dinheiro no tempo (taxa mínima de atratividade). A questão de definir qual taxa a ser empregada pode ser respondida por meio do seguinte exemplo (adaptado de [OLI82]):

Suponha-se que uma pessoa se encontre diante de uma oportunidade de investimento, para cuja concretização seja necessário tomar dinheiro emprestado de certo banco. Os juros pagos pelo empréstimo representarão, sob o ponto de vista do investidor, um ônus, o que deve ser entendido como o custo da utilização deste capital.

Naturalmente, a pessoa somente estará disposta a investir se a expectativa de ganhos, já deduzido o valor do investimento, for superior ao custo do capital. No caso em pauta, se o montante de juros pagos corresponderem a uma taxa de 40% ao ano, o custo do capital será expresso por este valor, e o investimento só será interessante se a taxa de rendimentos produzidos for superior a esse.

Tal fato identifica o custo do capital como sendo a rentabilidade mínima aceitável para qualquer aplicação, caracterizando então, uma base para aceitação ou rejeição de propostas de investimento. Essa taxa de juros é usualmente denominada como *taxa mínima de atratividade*.

4.3 CONSIDERAÇÕES

Em [FAV96] é apresentado um comparativo entre abordagens para análise de investimento em reuso, dentre elas VPL e Payback. Através da análise das características de cada método de análise de investimento é estabelecido que, para avaliação de investimento em reuso, o método VPL apresenta características mais adequadas e maior precisão na análise de retorno de investimento.

Partindo das conclusões de [FAV96], foi estabelecido que o modelo de análise de viabilidade de componentização de software deveria seguir as características dos métodos de VPL, considerando os fluxos de caixa em todo o ciclo de vida do componente. A incorporação de VPL ao modelo proposto é esclarecida na Seção 6.2.

5 MODELOS PARA ANÁLISE DE RETORNO DE INVESTIMENTO

A presente seção visa apresentar propostas encontradas na literatura de modelos de análise de retorno de investimento para criação de componentes. Após a apresentação do modelo proposto, retornaremos a esses trabalhos para ser estabelecida uma comparação entre essas propostas e o modelo proposto neste trabalho (Capítulo 8), dessa forma, destacando as contribuições acrescentadas pelo mesmo.

5.1 ABORDAGEM DE JEFFREY POULIN E JOSEPH CARUSO [POU93]

Na Seção 3.1.2 foram apresentadas as métricas do trabalho de [POU93] para avaliação dos custos de desenvolvimento e reutilização de componentes. Nesta seção, voltamos a este trabalho, agora para apresentar as métricas propostas pelo autor para análise de retorno de investimento em projetos com desenvolvimento baseado em componentes. Em [POU93], investimentos em DBC podem ser vistos de duas formas:

- ***Reuso ao Nível de Projeto:*** O reuso ao nível de projeto é caracterizado pela incorporação de preocupações relativas ao custo de desenvolvimento de componentes reutilizáveis, e como esse investimento será retornado com base em seus futuros reusos. Dessa forma, o reuso a nível de projeto abrange o mesmo universo referente ao modelo proposto neste trabalho.
- ***Reuso ao Nível Corporativo:*** O reuso a nível corporativo além de incorporar os custos/benefícios de reusos ao nível de projeto, também é composto pelos investimentos que uma organização deve realizar inicialmente para implantar um programa de reuso. Dentre esses custos podemos citar: custo de aquisição de um repositório de componentes, custo de promover o reuso dentro da empresa, custo de mudança dos seus atuais processos, dentre outros.

Como somente o Reuso ao Nível de Projeto apresenta o mesmo objetivo do modelo proposto, serão apresentadas exclusivamente as métricas relativas a este. Inicialmente,

[POU93] apresenta o cálculo de retorno de investimento como apresentado na Equação 18.

$$ROI = RCA + ORCA - ADC \quad (18)$$

Onde:

ROI: Retorno de investimento

CPR: Custo poupado pelo reuso no primeiro projeto

CPRD: Custo poupado pelo reuso nos demais reusos

CDC: Custo de desenvolvimento do componente

Custo poupado pelo reuso já foi previamente definido na Equação 10. O custo poupado pelo reuso nos demais reusos também é baseado na Equação 10. O custo de desenvolvimento do componente deve ser realizado como previamente definido na Equação 11.

5.2 ABORDAGEM DE RUTH MALAN E KEVIN WENTZEL [MAL93]

Em [MAL93] é apresentado um modelo para análise de custos e benefícios referentes ao reuso de componentes de software. Esse trabalho destaca-se por propor como traduzir a natureza qualitativa dos riscos para uma forma quantitativa, desta forma, integrando o risco no desenvolvimento do componente as métricas. A função de cálculo de retorno de investimento proposta em [MAL93] é apresentada na Equação 19.

$$ROI = \left[\sum_{i=1}^n (CNR_i - CRC_i) \times PR_i \right] - CDC + CER \quad (19)$$

Onde:

ROI: Retorno de investimento

CPR: Custo de não reutilização o componente na aplicação i

CPRD: Custo de reutilização o componente na aplicação i

CDC: Custo de desenvolvimento do componente

CER: Custos extras para reuso. Esta variável compreende os custos adicionais que a organização terá para implantar um programa de reuso

PR: probabilidade de reuso do componente na aplicação i

Apesar de [MAL93] propor uma forma de incorporar questões referentes a risco à análise de investimento, fica faltante na proposta a definição de um conjunto de possíveis riscos a serem considerados.

6 MODELO PROPOSTO

Embora o reuso de componentes possa apresentar inúmeras vantagens, são necessários componentes à disposição para poder se compor uma aplicação DBC. É consenso na literatura que o desenvolvimento de componentes reutilizáveis é mais custoso que o desenvolvimento de software sem focar em reuso. Esse custo extra enfatiza a importância de saber quando um componente irá retornar o investimento feito no seu desenvolvimento.

O presente trabalho apresenta uma proposta de modelo de análise de viabilidade que avalia quando é indicado o desenvolvimento de componentes reutilizáveis. Este modelo irá servir como ferramenta no auxílio da tomada de decisão de quando é financeiramente interessante para uma empresa a criação de um componente reutilizável.

O modelo de análise de viabilidade apresentado neste capítulo é baseado em grande parte na base teórica anteriormente apresentada. Partindo deste referencial teórico o modelo foi construído de forma incremental até a presente versão. A seguir são descritas as etapas percorridas na construção do mesmo.

Inicialmente, para a criação do modelo de análise de viabilidade, foram identificados os custos envolvidos no desenvolvimento de componentes e os fluxos de caixa dos projetos que reutilizam e não reutilizam componentes. Por fim, foi analisado como esses custos poderiam ser combinados de forma que fosse possível uma análise quantitativa do retorno sobre o investimento.

O modelo de análise de viabilidade mapeia três fontes de custos envolvidas no processo de desenvolvimento e reuso de um componente. Essas fontes são combinadas de forma a poder refletir o benefício (redução de custos), em termos monetários, do reuso de componentes. As fontes de custo citadas são:

- Custo de desenvolver um componente reutilizável;
- Custo do reuso deste componente: dada uma aplicação qualquer, qual o custo de se reutilizar este componente nesta aplicação; e
- Custo de não reutilizar este componente: dada uma aplicação qualquer, qual o custo de desenvolvimento das mesmas funcionalidades contidas no componente de uma forma tradicional, ou seja, sem reuso.

A Figura 6 ilustra dois cenários hipotéticos de desenvolvimento de software, um tradicional (sem reuso de componentes) e outro com reuso de um componente a ser desenvolvido. Pode ser observado que o desenvolvimento com componentes inicia com um custo extremamente elevado, pois para o primeiro reuso estão somados os custos de desenvolvimento e da primeira reutilização do componente.

Conforme o número de reusos aumenta, o desenvolvimento com componentes começa a ser viável, pois somente é necessário integrar o componente previamente desenvolvido. Em contrapartida, o custo de desenvolver sem componentes apresenta um crescimento linear, rentável até a segunda aplicação, após isto, o desenvolvimento com componentes se torna uma opção economicamente mais atrativa.

A identificação do momento em que a reutilização de componentes se torna rentável é vital para a tomada de decisão de quando componentizar. Esse retorno de investimento é visto na Figura 6 no cruzamento das linhas de reuso de componentes e sem reuso. Após a intersecção dessas linhas, a opção de reutilização de componentes começa a se tornar rentável.

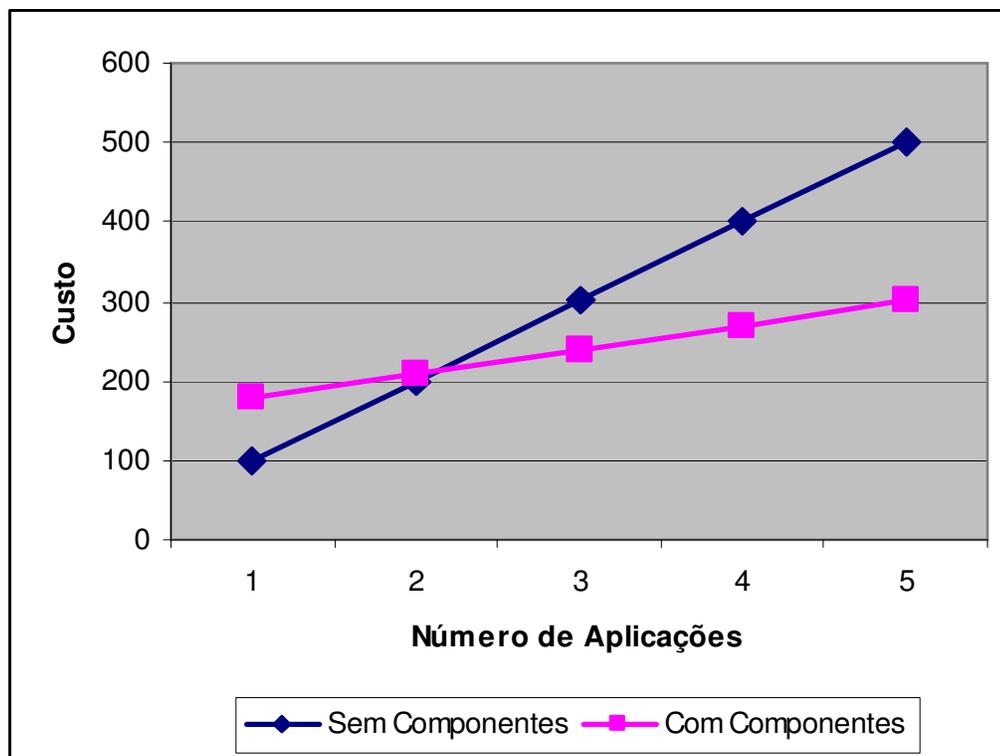


Figura 6. Retorno do investimento em componentização

Com base nesta combinação de custos, o modelo de análise de viabilidade foi criado. Inicialmente o modelo apresentava uma estrutura de *Payback* (técnica de análise de retorno de investimento apresentada na seção 4.1), posteriormente foi evoluído para VPL (baseado na técnica apresentada na seção 4.2) e ao final foram incorporadas características qualitativas visando permitir uma avaliação mais apurada dos cenários de reuso. Abaixo será apresentada a evolução do modelo e suas respectivas métricas.

6.1 ANÁLISE QUANTITATIVA BÁSICA (PAYBACK)

Com base na análise de investimento *Payback*, abordada na seção 4.1, uma primeira versão do modelo de análise de viabilidade foi elaborada. O relacionamento dos três custos anteriormente citados forma a base da análise de *Payback*. Esse relacionamento é expresso através da Equação 20. Ela cruza os custos de desenvolvimento com componentes, sem componentes e produção do artefato reutilizável. O número de aplicações (NRA) foi incluído à equação para que fosse possível a avaliação progressiva do retorno de investimento, conforme o número de projetos que reutilizam o componente.

$$CPR = \left[\sum_{i=1}^{NRA} (CNR - CRC) \right] - CDC \quad (20)$$

Onde:

CPR: Custo Poupado pelo reuso

CNR: Custo de não reutilizar o componente

NRA: Número de aplicações

CDC: Custo de desenvolvimento do componente

CRC: Custo de reuso do componente

O valor de CPR equivale ao prejuízo ou lucro em unidades monetárias do reuso. Para o cálculo das variáveis CNR, CRC e CDC foram utilizadas métricas baseadas nas apresentadas nos trabalhos citados na Seção 3.1. Algumas métricas requerem o uso de alguma técnica de medição de tamanho de software, como por exemplo, *Use Case Points* ou *Function Points*. O modelo proposto não apresenta nenhuma limitação referente à qual medida de software será utilizada, tornando assim esta opção de livre escolha do usuário do modelo. Na seqüência seguem as definições das métricas utilizadas para o cálculo de cada uma destas

variáveis.

6.1.1 Custo de Desenvolvimento do Componente (CDC)

O esforço requerido para se escrever um componente reutilizável é consideravelmente maior do que o de se desenvolver exatamente as mesmas funcionalidades em uma abordagem tradicional. O esforço adicional vem da necessidade de se desenvolver um componente com funcionalidades “genéricas”, que possam atender as necessidades de diversas aplicações, e demais esforços requeridos pelo modelo de componentes utilizado. Ao final desta fase o componente estará implementado e encapsulado.

O custo de desenvolvimento de componentes visa contabilizar todos os gastos referentes à criação do componente. Para o cálculo desta variável de custo, foi definida uma função adaptada da proposta de [DAG03]. A adaptação realizada foi a tradução do resultado da equação de esforço para financeiro.

Na Equação 21, é utilizada variável EE, que reflete o esforço extra despendido para o desenvolvimento de um componente reutilizável. Em [POU94] é proposto o valor de 1.5 e em [DAG03] 1.55. O ideal é refinar o valor desta variável através do histórico de desenvolvimento de componentes na empresa. Caso a empresa não possua este dado histórico é aconselhado o uso do valor 1.55, por ter sido levantado a partir de projetos mais recentes.

$$CDC = TF \times EE \times CD \quad (21)$$

Onde:

CDC: custo de desenvolvimento do componente

TF: tamanho (em alguma unidade de medida de software) das funcionalidades do componente

EE: esforço extra. Esta variável representa o esforço excedente de se criar componentes reutilizáveis. Seu valor pode ser definido com dados históricos da empresa.

CD: custo de desenvolvimento de uma unidade de medida de software. Este dado deve ser obtido através do histórico da empresa.

6.1.2 Custo de Reuso do Componente (CRC)

O desenvolvimento de software com componentes apresenta um conjunto de atividades relacionadas que levam à criação de uma aplicação através da composição de componentes pré-existentes. Toma-se como base que o componente já foi selecionado para

ser integrado à aplicação alvo, por isso, as atividades são consideradas a partir da sua integração. A Equação 22 representa a composição dos custos destas atividades.

$$CRC = TF \times TAC \times CD \quad (22)$$

Onde:

CRC: custo de reuso do componente

TF: tamanho (em alguma unidade de medida de software) das funcionalidades do componente

TAC: taxa de modificações do componente

CD: custo de desenvolvimento de uma unidade de medida de software. Este dado deve ser obtido através do histórico da empresa.

A Equação 22 foi adaptada a partir das equações 2, 3 e 4, apresentadas em [DAG03]. A adaptação realizada foi a unificação das três equações em uma, dessa forma adequando à natureza do modelo proposto que considera somente um componente por análise. Assim como apresentado anteriormente, o custo de reuso do componente irá variar conforme a quantidade de adaptações necessárias para reutilizar o componente (variável TAC da Equação 22). De forma sucinta pode-se citar que os três possíveis cenários são:

- Componentes que não necessitem ser adaptados para serem reutilizados irão possuir TAC igual a 0.2.
- Componentes que necessitem ser adaptados em 25% ou menos das suas funcionalidades apresentam TAC igual a 0.4.
- Componentes que necessitem ser adaptados em mais de 25% das suas funcionalidades apresentam TAC igual a 0.9.

A Equação 22 tem como resultado o custo de se reutilizar o componente em uma dada aplicação. O valor do TAC pode ser variado para se obter diversos cenários de reuso do componente.

6.1.3 Custo de não Reutilizar o Componente (CNR)

A variável CNR visa mapear os custos referentes ao desenvolvimento das mesmas funcionalidades do componente, só que seguindo uma abordagem tradicional, ou seja, sem reuso de componentes. Sua função (Equação 23) foi baseada no processo de desenvolvimento de software a partir da fase de implementação e reflete os mesmos princípios da Equação 13.

$$CNR = CD + CM \quad (23)$$

Onde:

CNR: Custo de não reutilizar o componente

CD: Custo de Desenvolvimento

CM: Custo de Manutenção

6.1.3.1 Custo de Desenvolvimento

A função de cálculo de custos referentes ao desenvolvimento é consideravelmente simples. É utilizado o tamanho das funcionalidades a serem implementadas e o custo de mão de obra. Essa função foi adaptada de [DAG03], apresentada neste trabalho na Equação 8. A Equação 24 apresenta esta função. A adaptação realizada foi a tradução do resultado da equação de esforço para financeiro.

$$CD = TF \times CD \quad (24)$$

Onde:

CD: custo de desenvolvimento.

TF: tamanho das funcionalidades a serem implementadas.

CD: custo de desenvolvimento de uma unidade de medida de software. Este dado deve ser obtido através do histórico da empresa

6.1.3.2 Custo de Manutenção

O custo de manutenção de uma aplicação que reutiliza componentes é mais baixo do que o de uma aplicação tradicional. Isso é explicado pelo fato de que componentes são artefatos de software de maior maturidade, logo possuindo menos erros. Para o modelo de análise de viabilidade proposto usamos a métrica apresentada em [POU93] para avaliar a redução do custo de manutenção (Equação 12). O custo de manutenção da aplicação é calculado basicamente através do uso de dados históricos da empresa. A Equação 25 reflete o custo referente à manutenção da porção da aplicação em que o componente se aplicaria.

$$CM = TF \times TE \times CMC \quad (25)$$

Onde:

CM: custo de manutenção

TF: tamanho (em alguma unidade de medida de software) das funcionalidades.

TE: taxa de erros por unidade de medida de software. Este dado deve ser obtido através do histórico da empresa.

CMC: Custo de manutenção de código, ou seja, custo médio de conserto de um bug. Este dado deve ser obtido através do histórico da empresa

6.1.4 Aplicação do Modelo

Para ilustrar de forma mais clara a aplicação das métricas será apresentado um exemplo de análise de retorno de investimento utilizando o modelo como apresentado até então. Esse mesmo exemplo também será utilizado em seções posteriores do presente trabalho conforme refinamentos no modelo forem apresentados.

Consideremos que uma empresa deseja aumentar o seu repositório de componentes. Para tal foi conduzido um processo de engenharia de domínio que resultou na identificação de três possíveis componentes reutilizáveis. Não estando certo do retorno de investimento relacionado a esses componentes o engenheiro de software da empresa decide executar uma análise de retorno de investimento.

O primeiro passo tomado pelo engenheiro é o de obter uma estimativa do tamanho dos componentes identificados. De posse dos artefatos gerados durante a engenharia de domínio e dos seus conhecimentos em *Use Case Points* o engenheiro levanta o tamanho dos componentes. O tamanho estimado para cada componente é apresentado na Tabela 3.

Tabela 3. Dados dos Componentes

Dado	Valor
Tamanho do Componente A	30 Use Case Points
Tamanho do Componente B	10 Use Case Points
Tamanho do Componente C	25 Use Case Points

De posse do modelo de análise de viabilidade de componentização de software o engenheiro começa a levantar as demais entradas do modelo. Baseado em parte nos dados do histórico de projetos da empresa, parte nas sugestões de índices do modelo o engenheiro de software alimenta o modelo. A Tabela 4 apresenta as demais entradas do modelo.

Tabela 4. Dados históricos

Dado	Valor
Custo de desenvolvimento por use case point	100 reais
Custo de manutenção por use case point	30
Esforço extra no desenvolvimento do componente	1.55
Taxa de erros por use case point	0.5

O gerente ao aplicar as métricas definiu que seria realizado um reuso por ano, durante três anos. As taxas de adaptação do Componente A seriam 10% no primeiro ano e 40% nos demais anos; o Componente B nunca precisaria ser adaptado; e o Componente C teria taxa de adaptação de 0% no primeiro ano, 20% no segundo ano e no terceiro 50%. A Tabela 5 apresenta os passos conduzidos pelo engenheiro de software para o cálculo do *payback* do componente. A Tabela 6 apresenta um quadro geral dos *paybacks* obtidos para os três componentes propostos.

Tabela 5. Passos para cálculo do Payback do componente A

Indicador	Métrica	
Custo de desenvolvimento do componente	$CDC = 30 \times 1.55 \times 100 = 4650$	
Custo de reuso do componente (reuso 1)	$CRC_1 = 30 \times 0.4 \times 100 = 1200$	
Custo de reuso do componente (reuso 2)	$CRC_2 = 30 \times 0.9 \times 100 = 2700$	
Custo de reuso do componente (reuso 3)	$CRC_3 = 30 \times 0.9 \times 100 = 2700$	
Custo de não reutilizar o componente	Custo de desenvolvimento	$CD = 30 \times 100 = 3000$
	Custo de Manutenção	$CM = 30 \times 0.5 \times 30 = 450$
	Custo total	$CNR = 3000 + 450 = 3450$
Custo poupado pelo reuso	$CPR_1 = \left[\begin{array}{l} (3450 - 1200) \\ + (3450 - 2700) \\ + (3450 - 2700) \end{array} \right] - 4650 = -900$	

Tabela 6. Análise de retorno de investimento com Payback

	Componente A	Componente B	Componente C
Investimento inicial	R\$ 4650	R\$ 1550	R\$ 3875
Ano	Rendimento por reuso	Rendimento por reuso	Rendimento por reuso
1	R\$ 2250	R\$ 950	R\$ 2375
2	R\$ 750	R\$ 950	R\$ 1875
3	R\$ 750	R\$ 950	R\$ 625
Payback	Não retornado em 3 anos	2 Anos	2 Anos

Baseado na análise Payback provida pelo modelo o engenheiro de software chega a conclusão que os componentes B e C são recomendados a serem de fatos componentizados, uma vez que ambos pagam o custo do seu desenvolvimento em dois anos. Como a idéia de Payback está mais ligada a uma data de ROI e não propriamente a valores monetários, ela não fornece uma base firme para que o engenheiro de software possa escolher somente um dos componentes para ser desenvolvido.

6.2 ANÁLISE QUANTITATIVA COMPOSTA (*VALOR PRESENTE LÍQUIDO*)

Quando as instâncias de reuso do componente ocorrem em janelas de tempo diferentes, os fluxos de caixa referentes devem ser tratados financeiramente de forma diferenciada. Essa característica é incorporada ao modelo através do uso da técnica de Valor Presente Líquido (VPL), abordagem apresentada na seção 4.2.

Para os custos anteriormente apresentados o VPL se aplica a duas variáveis: o CNR (custo de não reutilizar o componente) e o CRC (custo de reuso do componente). O VPL é aplicado a estas variáveis, pois os fluxos de caixa referentes a elas se dão em diferentes momentos do tempo, logo sendo afetadas pela desvalorização monetária implícita. A aplicação do VPL ao modelo é apresentada na Equação 26.

$$CPR = \left[\sum_{i=1}^{NRA} \left(\frac{CNR - CRC}{(1 + TMA)^{ano}} \right) \right] - CDC \quad (26)$$

Onde:

CPR: Custo Poupado pelo reuso

CNR: Custo de não reutilizar o componente

NRA: Número de aplicações

CDC: Custo de desenvolvimento do componente

CRC: Custo de reuso do componente

TMA: taxa mínima de atratividade

ano: ano de reuso do componente. Partindo do pressuposto que o ano de criação do componente é o ano zero.

A variável TMA deve seguir as mesmas definições apresentadas na Seção 4.2 para taxas mínimas de atratividade. A variável *ano* reflete o ano em que um dado projeto reutiliza o componente que está sendo avaliado, sendo que os anos devem começar a ser contados a partir de 1, ou seja, uma aplicação que reutiliza um componente no mesmo ano em que ele foi gerado tem a variável ano igual a 1.

Na presente versão do modelo de análise de viabilidade o engenheiro de software já pode conduzir uma análise quantitativa do retorno de investimento, diferentemente da versão Payback do modelo, desta forma podendo diferenciar, por exemplo, retornos de investimento de componentes com o mesmo Payback. Outra melhoria incorporada ao modelo é a análise utilizando valores adaptados a uma janela de tempo presente, desta forma trazendo mais precisão aos resultados fornecidos pelo modelo, aumentando à confiabilidade do mesmo para a tomada de decisão.

6.2.1 Aplicação do Modelo

O exemplo apresentado na Seção 6.1.4 será novamente utilizado, desta vez para demonstrar a utilização do modelo integrando a técnica de VPL. Para aplicar a nova versão do modelo de análise de viabilidade de componentização de software o engenheiro de software deve estabelecer a taxa mínima de atratividade. Para o contexto do nosso exemplo será utilizada uma taxa de 10% ao ano, todos os demais dados de entrada do modelo serão utilizados como apresentados na Seção 6.1.4. Partindo dos cálculos apresentados da Tabela 5, o engenheiro de software aplica a técnica de valor presente líquido. Dessa forma, somente será recalculado o custo poupado pelo reuso (CPR) como apresentado abaixo.

$$CPR = \left[\frac{(3450 - 1200)}{(1 + 0.1)^1} + \frac{(3450 - 2700)}{(1 + 0.1)^2} + \frac{(3450 - 2700)}{(1 + 0.1)^3} \right] - 4650 = -1421.24$$

Após aplicar as entradas ao modelo o engenheiro de software obteve os resultados apresentados na Tabela 7.

Tabela 7. Análise de retorno de investimento com VPL

	Componente A		Componente B		Componente C	
Investimento inicial	R\$ 4650		R\$ 1550		R\$ 3875	
Ano	Rendimento por reuso	Fluxos de caixa	Rendimento por reuso	Fluxos de caixa	Rendimento por reuso	Fluxos de caixa
1	R\$ 2045.45	-2604.55	R\$ 863.63	-686.36	R\$ 2159.09	-1715.9
2	R\$ 619.83	-1984.71	R\$ 785.12	98.76	R\$ 1549.58	-166.32
3	R\$ 563.48	-1421.22	R\$ 713.74	812.5	R\$ 469.57	303.24

Fazendo uso da nova versão do modelo o retorno de investimento pode ser visto como fluxos de entrada e saída de caixa. Os fluxos de caixa para desenvolvimento de componentes iniciam com um fluxo de saída, que é o investimento inicial para criação do componente. Gradualmente, a cada reuso, ocorrem fluxos de entrada, que são decorrentes do custo poupado pela integração do componente. Se compararmos os rendimentos por reuso apresentados na Tabela 6, pode-se notar o decréscimo nos rendimentos dos reusos, acarretados pela aplicação da taxa mínima de atratividade. Esse decréscimo refina os valores para comparação de análise de investimento.

Para avaliar as alternativas de reuso nesta versão do modelo, devemos avaliar o fluxo de caixa final. No caso do exemplo proposto, deve-se analisar o fluxo de caixa do ano três. Novamente os componentes B e C conseguem retornar o investimento realizado nas suas criações. Na versão VPL do modelo podemos diferenciar os benefícios de criação do componente B e C em termos financeiros. Como o componente B apresenta maior fluxo de caixa final, caso a empresa somente pudesse criar um componente nesse momento, o mais indicado seria o B por gerar maiores lucros.

6.3 ANÁLISE QUALITATIVA

Na Seção 3.2 foi apresentado um grupo de riscos encontrados na literatura referentes à obsolescência do componente de software. O modelo proposto incorporou a sua estrutura todos os riscos levantados nessa seção, para, desta forma, prover uma maior liberdade ao engenheiro de software na criação de cenários que melhor se encaixem a sua realidade.

Assim como na proposta de [EZR02], o modelo vai integrar os riscos como uma série de perguntas que o engenheiro deve responder no momento da tomada de decisão sobre a criação de um componente. Abaixo são apresentadas as perguntas a serem respondidas para cada um dos riscos levantados:

- Manutenção: Qual a probabilidade de ainda ser provida manutenção ao componente na aplicação X?
- Documentação: Qual a probabilidade da documentação do componente suprir as necessidades de reuso na aplicação X?
- Performance: Qual a probabilidade do componente ainda suprir as necessidades de performance das aplicações na aplicação X?
- Obsolescência Funcional: Qual a probabilidade das funcionalidades providas no componente ainda serem adequadas na aplicação X?
- Obsolescência Técnica: Qual a probabilidade da arquitetura da qual o componente depende ainda será utilizada na aplicação X?

A próxima sessão apresentada como os riscos identificados foram incorporados de forma quantitativa ao conjunto de métricas do modelo.

6.3.1 Quantificação de Valores Qualitativos

Em [MAL93] é levantada a importância de se incorporar os riscos de um componente não vir a ser reutilizado em futuras aplicações devido a incertezas decorrentes da passagem do tempo. Nesse mesmo trabalho é apresentado um conjunto de métricas para se determinar o retorno de investimento de criação de componentes. As diferenças desse trabalho frente ao aqui apresentado são que nas métricas de [MAL93] é necessário saber se antes os dados de todas as aplicações que irão reutilizar o componente; também não são definidos quais os riscos inerentes ao desenvolvimento de componentes. Apesar de não apresentar quais são os riscos, [MAL93] sugere como incorporá-los às métricas. Essa sugestão foi incorporada ao conjunto de métricas apresentado neste trabalho, e é representada pela variável PR na Equação 27.

$$CPR = \left[\sum_{i=1}^{NRA} \left(\frac{CNR - CRC}{(1 + TMA)^{ano}} \right) \times PR_i \right] - CDC \quad (27)$$

Onde:

CPR: Custo Poucado pelo reuso

CNR: Custo de não reutilizar o componente

NRA: Número de aplicações

CDC: Custo de desenvolvimento do componente

CRC: Custo de reuso do componente

TMA: taxa mínima de atratividade

ano: ano de reuso do componente. Partindo do pressuposto que o ano de criação do componente é o ano zero.

PR: Probabilidade de reutilização do componente na aplicação i.

A probabilidade de reutilização do componente poderá ser obtida através da média ponderada das probabilidades dos fatores atenderem as necessidades da aplicação multiplicadas pelos seus respectivos pesos. O conceito de peso é incorporado ao modelo para possibilitar a priorização de fatores qualitativos, dessa forma, permitindo ao engenheiro de software avaliar diversos cenários de reuso. É possível que cada aplicação ao reutilizar o componente possua uma diferente probabilidade de reutilização. Isto se dá, pois os reusos ocorrem em diferentes momentos do tempo, conseqüentemente, possuindo diferentes taxas de obsolescência. Os possíveis pesos são apresentados na Tabela 8.

Tabela 8. Pesos dos fatores de ajuste

Peso	Descrição
1	Não se aplica
2	Desnecessária
3	Desejável
4	Necessária
5	Fundamental

6.3.2 Aplicação do Modelo

Para exemplificar o uso dos fatores qualitativos será continuado o exemplo previamente apresentado na seção 6.2.1. Para refinar os resultados financeiros obtidos através da aplicação de valor presente líquido o engenheiro de software decide aplicar as variáveis qualitativas apresentadas na Seção 6.3. Para tal, o engenheiro prospecta os riscos de cada componente baseado nas suas tecnologias e naturezas. A Tabela 9 apresenta os riscos definidos para o componente A, a Tabela 10 apresenta os do componente B e a Tabela 11 apresenta os do componente C.

Tabela 9. Riscos do componente A

	Ano 1	Ano 2	Ano 3
Manutenção	100%	90%	80%
Peso da Manutenção	Necessária	Necessária	Necessária
Documentação	100%	100%	100%
Peso da Documentação	Desejável	Desejável	Desejável
Performance	100%	90%	80%
Peso da Performance	Fundamental	Fundamental	Fundamental
Obsolescência Funcional	100%	90%	80%
Peso da Obsolescência Funcional	Fundamental	Fundamental	Fundamental
Obsolescência Técnica	100%	100%	100%
Peso da Obsolescência Técnica	Fundamental	Fundamental	Fundamental

Tabela 10. Riscos do componente B

	Ano 1	Ano 2	Ano 3
Manutenção	100%	100%	100%
Peso da Manutenção	Fundamental	Fundamental	Fundamental
Documentação	100%	100%	100%
Peso da Documentação	Desnecessária	Desnecessária	Desnecessária
Performance	100%	80%	70%
Peso da Performance	Fundamental	Fundamental	Fundamental
Obsolescência Funcional	100%	100%	80%
Peso da Obsolescência Funcional	Fundamental	Fundamental	Fundamental
Obsolescência Técnica	100%	70%	70%
Peso da Obsolescência Técnica	Não se aplica	Não se aplica	Não se aplica

Tabela 11. Riscos do componente C

	Ano 1	Ano 2	Ano 3
Manutenção	80%	60%	50%
Peso da Manutenção	Necessária	Necessária	Necessária
Documentação	90%	90%	90%
Peso da Documentação	Fundamental	Fundamental	Fundamental
Performance	80%	70%	50%
Peso da Performance	Desejável	Desejável	Desejável
Obsolescência Funcional	90%	90%	80%
Peso da Obsolescência Funcional	Fundamental	Fundamental	Fundamental
Obsolescência Técnica	90%	90%	80%
Peso da Obsolescência Técnica	Fundamental	Fundamental	Fundamental

Inicialmente, o engenheiro de software decide calcular a probabilidade de reuso para os três reusos do componente A. A partir dos dados apresentados na Tabela 9 o engenheiro conduz os cálculos apresentados na tabela 12.

Tabela 12. Cálculo da probabilidade de reuso do componente A

Indicador	Métrica
Probabilidade de reuso para a primeira aplicação	$PR_1 = \frac{4 \times 1 + 3 \times 1 + 5 \times 1 + 5 \times 1 + 5 \times 1}{22} = 1$
Probabilidade de reuso para a segunda aplicação	$PR_1 = \frac{4 \times 0.9 + 3 \times 1 + 5 \times 0.9 + 5 \times 0.9 + 5 \times 1}{22} = 0.93$
Probabilidade de reuso para a terceira aplicação	$PR_1 = \frac{4 \times 0.8 + 3 \times 1 + 5 \times 0.8 + 5 \times 0.8 + 5 \times 1}{22} = 0.87$

De posse das probabilidades de reuso do componente A, o engenheiro de software calcula o retorno de investimento como definido na Equação 27. Abaixo é apresentado o cálculo realizado pelo engenheiro.

$$CPR = \left\{ \left[\frac{(3450 - 1200)}{(1 + 0.1)^1} \right] \times 1 + \left[\frac{(3450 - 2700)}{(1 + 0.1)^2} \right] \times 0.93 + \left[\frac{(3450 - 2700)}{(1 + 0.1)^3} \right] \times 0.87 \right\} - 4650 = -1532.38$$

A seguir o engenheiro de software realiza os mesmos cálculos para os demais componentes. Ao aplicar o modelo com as variáveis qualitativas o engenheiro de software obtém um resultado diferente das análises previamente realizadas. Nessa nova análise, o único componente que foi capaz de retornar o investimento realizado na sua criação foi o componente B. O excesso de riscos e incertezas relacionados ao componente C fez o seu retorno de investimento decair a um patamar negativo, dessa forma, se tornando um investimento que acarretaria prejuízos a empresa. Os dados obtidos através da nova aplicação do modelo são apresentados na Tabela 13.

Tabela 13. Análise de retorno de investimento com fatores qualitativos

	Componente A		Componente B		Componente C	
Investimento inicial	R\$ 4650		R\$ 1550		R\$ 3875	
Ano	Rendimento por reuso	Fluxos de caixa	Rendimento por reuso	Fluxos de caixa	Rendimento por reuso	Fluxos de caixa
1	R\$ 2045.46	-2604.54	R\$ 863.64	-686.36	R\$ 1874.49	-2000.51
2	R\$ 580.39	-2024.15	R\$ 644.31	42.05	R\$ 1267.84	-732.67
3	R\$ 491.77	-1532.38	R\$ 602.72	644.77	R\$ 341.51	-391.16

Com base nos exemplos apresentados foi possível demonstrar que através do uso das técnicas de valor presente líquido e da adição de fatores qualitativos ao modelo, o mesmo pôde refinar os valores de retorno de investimento. Dessa forma, pode servir como ferramenta de auxílio a engenheiros de software incumbidos de realizar a tomada de decisão na criação de componentes reutilizáveis.

Os dados apresentados na Tabela 13 podem ser vistos de forma visual na Figura 7. O eixo Y representa o fluxo de caixa, o eixo X apresenta uma escala de tempo, onde cada nodo das linhas representa 1 (um) reuso em um dado momento do tempo. O gráfico apresentado na Figura 7, foi gerado pela ferramenta implementada para suportar a aplicação do modelo. A ferramenta será apresentada em mais detalhes na Seção 6.3.3.

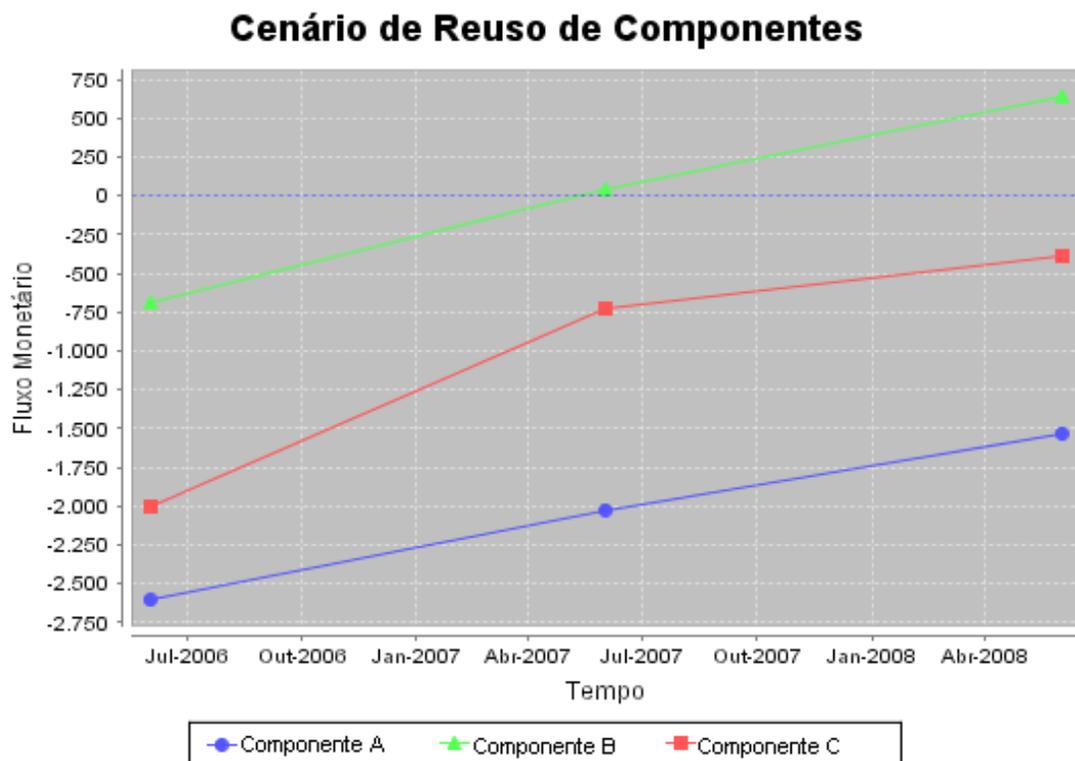


Figura 7. Cenário de reuso de componentes

6.3.3 Automatização do Modelo

Para auxiliar a aplicação do modelo, bem como para facilitar a criação e comparação de diversos cenários, foi desenvolvida uma ferramenta aderente as métricas propostas no modelo. Engenheiros de software que fizerem uso da ferramenta poderão criar e comparar diversos cenários de desenvolvimento de componentes. A geração de cenários é especialmente utilizada para comparação de instâncias de reuso complexas, onde existe alto grau de incerteza nas entradas fornecidas ao modelo.

Para utilizar a ferramenta o engenheiro de software deve inicialmente alimentar o software com as entradas do modelo. As entradas estão divididas na forma de abas na ferramenta, desta forma, facilitando o agrupamento de dados relacionados. Abaixo são apresentados os grupos de entradas requeridas pela aplicação.

- **Dados do Componente:** Esta aba agrupa os dados necessários sobre o componente a ser desenvolvido. Como o modelo tem como objetivo ser utilizado na prospecção de possíveis artefatos reutilizáveis, poucos são os dados disponíveis sobre o componente. A Figura 8 apresenta esse grupo de dados na ferramenta.
 - **Tamanho das funcionalidades:** a única informação do componente requerida pelo modelo é o tamanho de suas funcionalidades. Esse tamanho pode ser fornecido em qualquer unidade de medida de software.

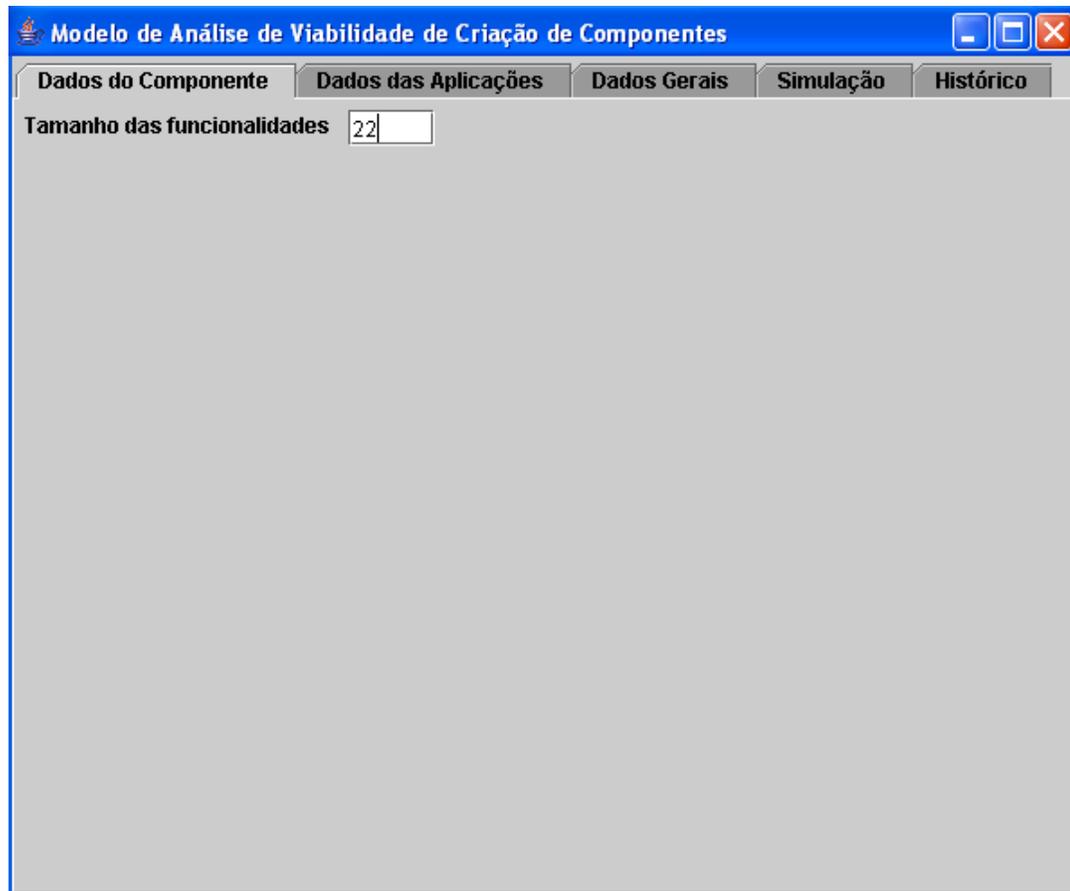
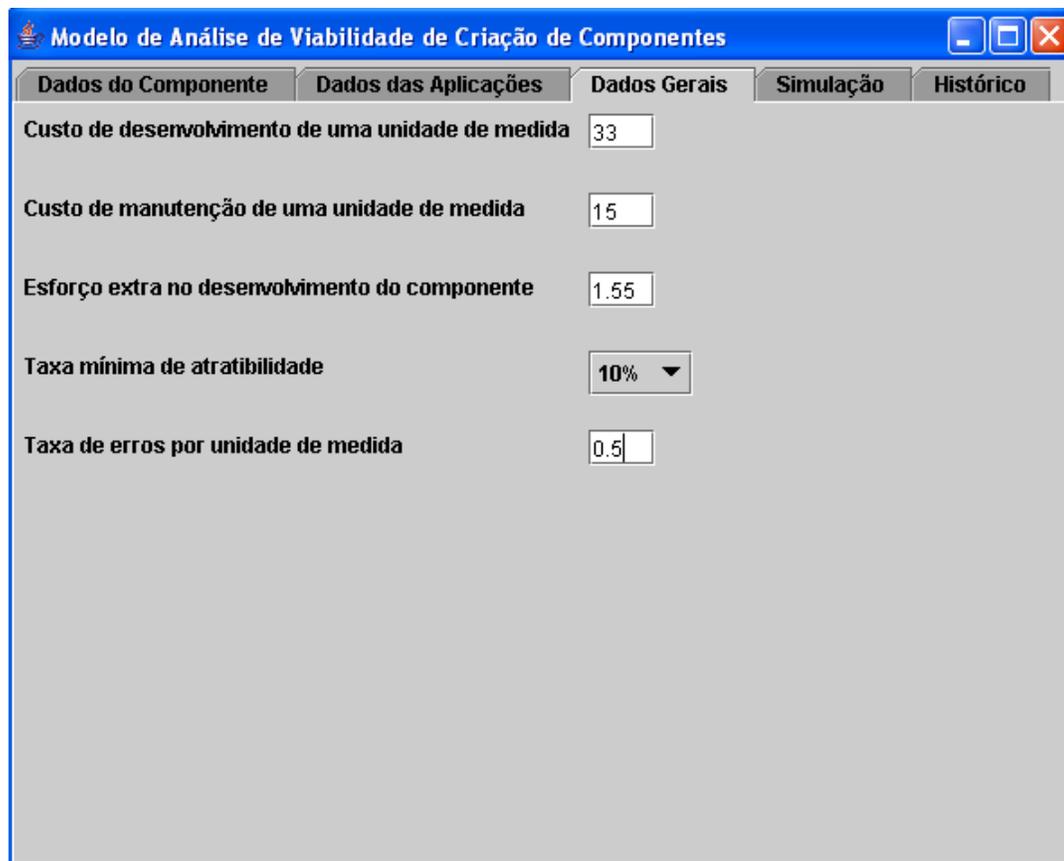


Figura 8. Ferramenta de aplicação do modelo (dados do componente)

- **Dados das Aplicações:** A aba de Dados das Aplicações é onde devem ser informados os possíveis reusos de um dado componente, apresentada na Figura 9. Essa é a aba onde o engenheiro de software poderá estabelecer diversos cenários de reuso do componente utilizando principalmente os fatores qualitativos anteriormente estabelecidos.
 - **Ano de Reuso:** Campo que contém o ano de reuso do componente. Caso o componente seja reutilizado no mesmo ano de sua criação deve ser colocado Ano de Reuso com valor 1, sendo incrementado em uma unidade a cada ano decorrente.
 - **Taxa de Adaptação:** A variável Taxa de Adaptação deve ser estabelecida conforme o percentual das funcionalidades do componente que tiveram de ser adaptadas para reutilizar o mesmo.

- Manutenção: A variável Manutenção se refere à probabilidade da disponibilidade de manutenção para o componente no dado ano de reuso. Sendo assim, um componente que terá manutenção integral em um dado ano terá a variável Manutenção com valor de 100%.
- Peso da Manutenção: Os fatores qualitativos podem possuir diferentes importâncias para diferentes componentes. Para tal foi criada uma escala de pesos na qual o engenheiro de software poderá estabelecer importâncias para os diferentes fatores qualitativos. A escala pode assumir os valores: Não se Aplica, Desnecessária, Desejável, Necessária e Fundamental.
- Documentação: A variável Documentação se refere à probabilidade da documentação estar atualizada e disponível em um dado ano de reuso.
- Peso da Documentação: Segue a mesma lógica da variável de Peso da Manutenção.
- Performance: A variável Performance se refere à probabilidade da performance das funcionalidades providas pelo componente ainda atenderem as necessidades da aplicação que o reutiliza em um dado ano.
- Peso da Performance: Segue a mesma lógica da variável de Peso da Manutenção.
- Obsolescência Funcional: A variável Obsolescência Funcional se refere à probabilidade do componente ainda prover serviços que funcionalmente atendam as necessidades da aplicação.
- Peso da Obsolescência Funcional: Segue a mesma lógica da variável de Peso da Manutenção.
- Obsolescência Técnica: A variável Obsolescência Técnica se refere à probabilidade do componente ainda possuir suporte pelas plataformas e tecnologias do qual o mesmo depende.
- Peso da Obsolescência Técnica: Segue a mesma lógica da variável de Peso da Manutenção.

- **Dados Gerais:** A aba de dados gerais agrupa os dados referentes à organização que faz uso do modelo, parte dos dados necessários devem ser extraídos do histórico de projetos passados da empresa. A aba de Dados Gerais é apresentada na Figura 10.
 - **Custo de Desenvolvimento de uma Unidade de Medida:** Como o próprio nome já diz, essa variável deve ser alimentada com o custo de desenvolvimento de uma unidade de medida, por exemplo, uma dada empresa tem o custo de desenvolvimento de 100 reais por ponto de função.
 - **Custo de Manutenção de uma Unidade de Medida:** Similar a variável de custo de desenvolvimento só que relacionada ao custo de manutenção.
 - **Esforço Extra no Desenvolvimento do Componente:** Essa variável deve ser alimentada com a taxa de esforço extra para se desenvolver um grupo de funcionalidades de forma a se tornarem reutilizáveis frente a serem desenvolvidas de forma tradicional. A ferramenta propõe por padrão o valor de 1.55. Esse valor foi previamente discutido na Seção 6.1.1.
 - **Taxa Mínima de Atratividade:** A ferramenta provê uma escala de valores para se estabelecer o valor dessa variável, cuja natureza já foi discutida na Seção 4.2.1.
 - **Taxa de Erros por Unidade de Medida:** Esta variável deve conter uma média de erros por unidade de medida após uma dada aplicação ser posta em produção. Essa variável deve ser alimentada com base nos dados do histórico de projetos da empresa.



The image shows a software window titled "Modelo de Análise de Viabilidade de Criação de Componentes". The window has a blue title bar with standard Windows window controls (minimize, maximize, close). Below the title bar, there are five tabs: "Dados do Componente", "Dados das Aplicações", "Dados Gerais", "Simulação", and "Histórico". The "Dados Gerais" tab is currently selected. The main area of the window is a light gray form with the following fields:

Label	Value
Custo de desenvolvimento de uma unidade de medida	33
Custo de manutenção de uma unidade de medida	15
Esforço extra no desenvolvimento do componente	1.55
Taxa mínima de atribibilidade	10% ▼
Taxa de erros por unidade de medida	0.5

Figura 10. Ferramenta de aplicação do modelo (dados gerais)

Após alimentar o modelo, a ferramenta permite que o engenheiro de software verifique de forma visual o retorno de investimento para o cenário estabelecido, como apresentado na Figura 11. Cada nodo no gráfico representa um dos reusos definidos na aba “Dados das Aplicações”, o eixo X apresenta a passagem do tempo e o Y o fluxo de caixa do retorno de investimento. Dessa forma, quando um dos nodos superar o valor 0 (zero) no eixo Y, significa que o investimento foi retornado.

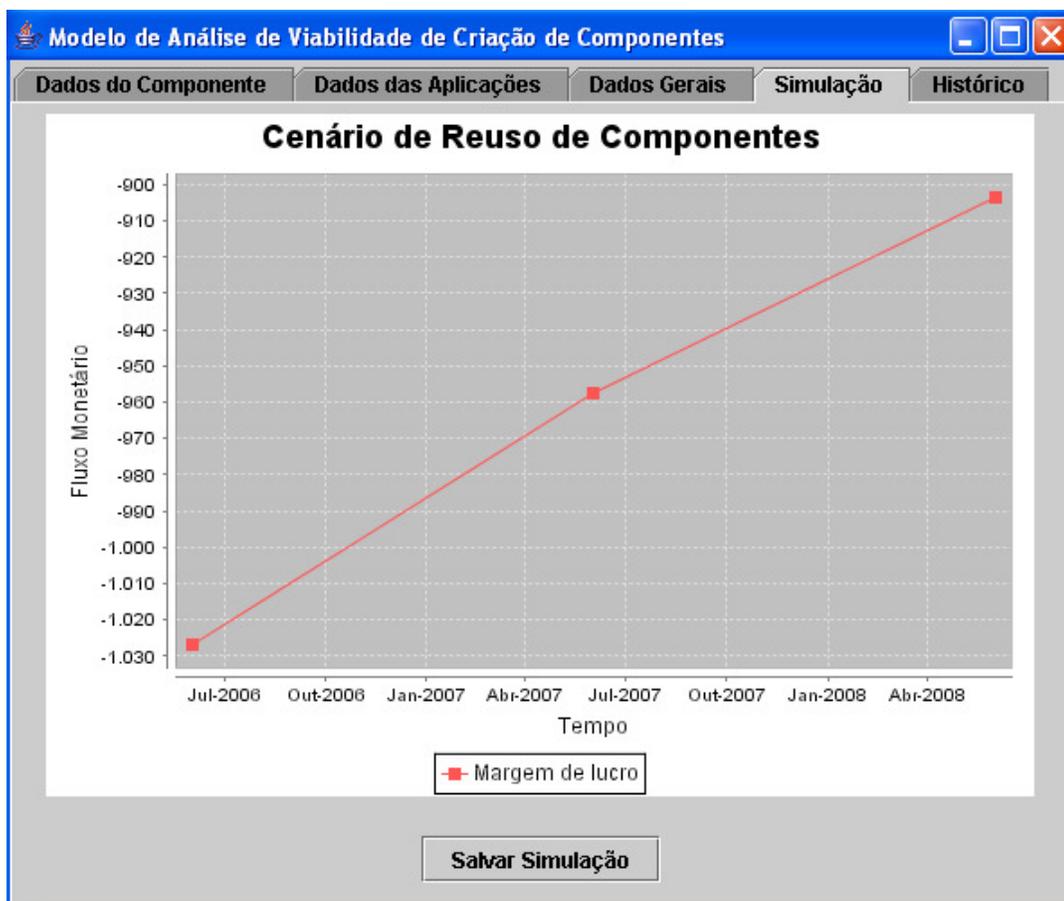


Figura 11. Ferramenta de aplicação do modelo (simulação de retorno de investimento)

Para facilitar a avaliação de diversos cenários à aplicação permite ao engenheiro de software salvar diversos cenários. Os mesmos podem posteriormente serem comparados na aba “Histórico”, apresentado na Figura 12.

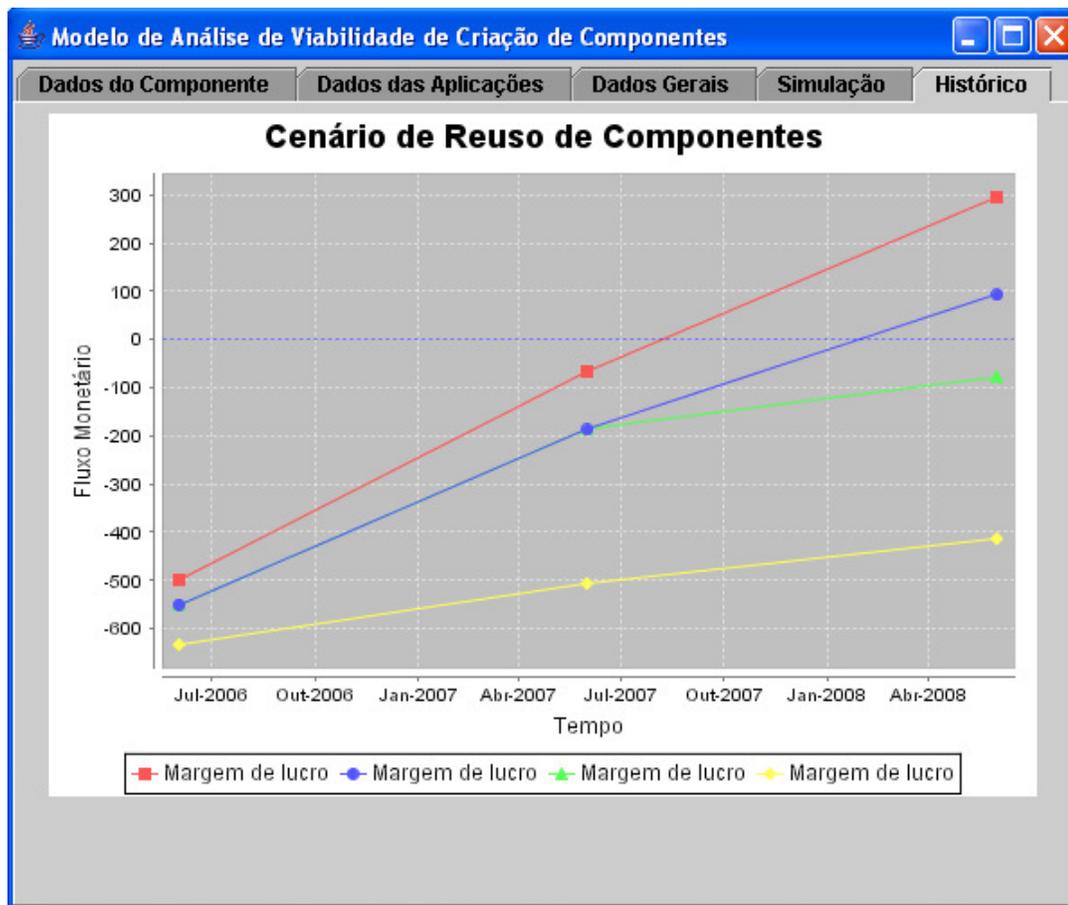


Figura 12. Ferramenta de aplicação do modelo (histórico de simulações)

7 AVALIAÇÃO DO MODELO

Partindo do modelo descrito anteriormente, foi realizada uma avaliação de campo para observar a importância e validade da dada proposta em um ambiente real de desenvolvimento de software. A avaliação de campo procurou observar a percepção dos especialistas das empresas, quanto à adequação do modelo de análise de viabilidade de componentização de software às reais necessidades da empresa. O instrumento de coleta de dados utilizado foi um questionário com questões fechadas que dependendo da resposta poderiam ser justificadas (Anexo I).

O cenário ideal de avaliação do modelo seria aplicá-lo repetidamente em diversos projetos e componentes, mas devido a limitações de tempo, optou-se por coletar a opinião de profissionais em diferentes empresas. Sendo assim, foi possível capturar o sentimento dos mesmos quanto à validade do uso do modelo na prospecção de componentes.

7.1 CARACTERIZAÇÃO DAS ORGANIZAÇÕES

A avaliação de campo foi desenvolvida nas unidades de desenvolvimento de software de duas organizações, uma de grande e outra de médio porte. A de médio porte se caracteriza por desenvolver sistemas encomendados por terceiros, enquanto a de grande porte faz parte de uma *holding* multinacional, cujo objetivo é desenvolver os sistemas requeridos pelas demais empresas do grupo. Ambas as empresas estão em fase de implantação de programas de reuso de componentes de software, assim sendo, sujeitas a tomada de decisão de criação de componentes de software.

7.2 ROTEIRO

Inicialmente foram realizados contatos com as empresas visando obter permissão para realização da pesquisa de campo. A seguir, foi elaborado o instrumento de coleta de dados e decidido quem seriam os respondentes.

O questionário foi submetido a uma validação de face e conteúdo. Essa foi realizada por dois professores da PUCRS (Pontifícia Universidade Católica do Rio Grande do Sul), ambos doutores em Ciência da Computação, os quais fizeram críticas e sugestões ao questionário possibilitando seu refinamento. A seguir, o questionário foi submetido a um pré-teste por dois respondentes, os quais não fizeram parte da análise dos dados, visando descobrir erros, eliminar dúvidas e ambigüidades nas questões do questionário. Assim, o questionário foi aplicado a profissionais ligados ao programa de reuso juntamente com um formulário de avaliação. Os resultados do pré-teste permitiram optar por formulações mais apropriadas das questões para o objetivo do estudo.

Após ajustado o questionário com os resultados do pré-teste, este foi aplicado aos respondentes definidos. Para que os entrevistados tomassem conhecimento do modelo proposto e pudessem responder ao questionário, foi apresentada a eles a ferramenta desenvolvida para aplicação do modelo. Permitindo assim, que os respondentes criassem diversos cenários de reuso e tomassem conhecimento do funcionamento do modelo proposto. Por fim, os questionários, devidamente respondidos, foram recolhidos para análise e obtenção dos resultados.

7.3 CARACTERIZAÇÃO DOS RESPONDENTES

Como o objetivo da aplicação do questionário era o de capturar as impressões de especialistas sobre a confiança no modelo para auxílio à tomada de decisão, foram selecionados profissionais diretamente relacionados às iniciativas de reuso de componentes das empresas. Foram selecionados dois profissionais de cada empresa para responderem o questionário, dentre eles dois gerentes (um com titulação de Especialista e outro Doutor) e dois projetistas (ambos com titulação de Mestre).

7.4 RESULTADOS

Para avaliação do modelo de análise de viabilidade de desenvolvimento de componentes proposto foi aplicado a um grupo de profissionais, da iniciativa privada, ligados a programas de reuso, o questionário apresentado no Apêndice I. Para cada uma das questões presentes no questionário serão apresentadas as respostas e observações levantadas pelos respondentes, seguidas de ponderações sobre as colocações apresentadas.

Quando perguntado aos respondentes “O modelo de análise de viabilidade de componentes auxilia na decisão de componentização de software?” as respostas obtidas foram as seguintes:

- Para 25% dos respondentes o modelo proposto auxilia na decisão de componentização de software.
- Para 75% dos respondentes o modelo auxilia parcialmente. Abaixo são apresentadas as justificativas colocadas como motivo para que o modelo não tenha auxiliado integralmente no processo de tomada de decisão.
 - Para um dos respondentes o modelo não se adere integralmente ao modelo de negócios da empresa, uma vez que a empresa desenvolve software para terceiros, dessa forma, dificultando a criação de cenários de reuso do componente.
 - Para empresas com este perfil, certamente a tarefa de previsão de reuso é dificultada. No entanto, componentes de infraestrutura, e componentes horizontais em geral, podem ser facilmente analisados no modelo, devido à larga aplicabilidade de suas funcionalidades, como por exemplo, um componente de *pool* de conexão para banco de dados.
 - Para dois respondentes o modelo auxilia na tomada de decisão, mas algumas das métricas propostas no modelo não se aplicam a realidade da empresa.
 - Os entrevistados colocam que na empresa em que trabalham os benefícios do reuso de componentes é encarado como ganho de produtividade dentro do projeto, e não, como redução de custos. O modelo proposto trabalha exclusivamente sobre ROI na prospecção de benefícios econômicos, sendo assim, a análise de redução de esforço em projetos foge do escopo do modelo proposto.

Posteriormente foi perguntado aos respondentes “Você julga que a possibilidade de geração de diferentes cenários através da parametrização das taxas de adaptabilidade ajuda na análise de viabilidade para criação de componentes?”. Abaixo é apresentado o retorno obtido.

- Para 75% dos respondentes a geração de diferentes cenários através da parametrização das taxas de adaptabilidade pode ajudar na análise de viabilidade para criação de componentes.
- Para 25% dos respondentes o modelo auxilia parcialmente.
 - O entrevistado coloca que os componentes desenvolvidos pela empresa em que trabalha são modificados a cada nova necessidade de uma aplicação que o reutiliza, ganhando uma nova versão.
 - Essa prática é desaconselhável para componentes de software, pois incrementar o componente a cada nova necessidade, pode incorporar ao mesmo funcionalidades não compartilhadas por outras aplicações. Dessa forma, descaracterizando a coesão do componente, incrementando a sua complexidade e possivelmente aumentando o risco de geração incompatibilidades entre o componente e outras aplicações que o tenham reutilizado previamente.

Quando perguntado “Você julga que a possibilidade de geração de diferentes cenários através da distribuição de custos em diferentes projetos ajuda na análise de viabilidade para criação de componentes?” o resultado obtido foi o seguinte:

- Para 100% dos respondentes a possibilidade de geração de diferentes cenários através da distribuição de custos em diferentes projetos ajuda na análise de viabilidade para criação de componentes.

A seguir, foi perguntado aos respondentes “Você julga que as variáveis de entrada do modelo são suficientes para estipular custos no desenvolvimento e reuso de componentes?”. Abaixo é apresentado o retorno obtido.

- Para 25% dos respondentes as variáveis quantitativas de entrada do modelo são suficientes para estipular custos no desenvolvimento e reuso de componentes.
- Para 75% dos respondentes as variáveis quantitativas de entrada do modelo atendem parcialmente a tarefa de estipular custos no desenvolvimento e reuso de componentes. A seguir são apresentadas as justificativas colocadas pelos

respondentes.

- Um dos respondentes propõe a apresentação de mais dados sobre a análise de retorno de investimento.
 - As propostas do entrevistado foram incorporadas à ferramenta.
- Para dois respondentes o modelo proposto deveria propor métricas para comprovação da acuidade das estimativas propostas.
 - O modelo propõe um modelo para análise de viabilidade para criação de componentes, e não análise dos benefícios do reuso em um momento pós-reutilização, tais métricas fogem do escopo do modelo proposto. Tais métricas podem ser encontradas em trabalhos como [POU93].

Posteriormente foi perguntado aos respondentes “Você julga que as variáveis qualitativas do modelo (Manutenção, Documentação, Performance, Obsolescência Funcional e Obsolescência Técnica) são suficientes para avaliação do ciclo de vida do componente?”. Abaixo é apresentado o retorno obtido.

- Para 75% dos respondentes variáveis qualitativas do modelo são suficientes para análise de viabilidade para criação de componentes.
- Para 25% dos respondentes as variáveis qualitativas suprem parcialmente a avaliação do ciclo de vida do componente.
 - O respondente coloca o seguinte: “Estimar estas variáveis e depois medir o realizado pelo projeto é extremamente difícil. Os efeitos das variáveis dependem do tipo de componente (comprado ou desenvolvido) e da estratégia de mudança (mudar o componente ou a aplicação)”.
 - Certamente os efeitos das variáveis propostas variam conforme o tipo de componente, por esta razão, o modelo possibilita ao engenheiro de software a possibilidade de jogar com diferentes valores de suporte a cada uma das variáveis propostas. Quanto à estratégia de mudança, o modelo também incorpora esta característica através da taxa de adaptação das funcionalidades do componente.

Após a contabilização e análise das respostas dos questionários pode-se verificar que o modelo proposto auxiliou na avaliação de cenários de desenvolvimento de componentes propostos pelos entrevistados. O modelo não sofreu avaliações negativas quanto a sua aplicabilidade e auxílio à tomada de decisão. As observações colocadas pelos entrevistados que não estavam fora do contexto do modelo proposto foram integradas ao mesmo. Os resultados obtidos através da aplicação do questionário possibilitaram a avaliação e evolução do modelo proposto.

8 CONSIDERAÇÕES FINAIS

Desenvolver software com qualidade e de forma produtiva são preocupações evidentes nas organizações de desenvolvimento de software atuais. O DBC surge como uma abordagem interessante, pois permite desenvolver sistemas de software através de componentes previamente desenvolvidos para o reuso. Dentre as vantagens oferecidas, inclui a melhoria no gerenciamento da complexidade do sistema, o aumento da produtividade, melhora na qualidade, um maior grau de consistência e um alto índice de usabilidade [BRO00]. Porém para usufruir de tais vantagens é necessária uma sistemática abordagem para DBC em nível de processos e tecnologia, pois DBC ainda apresenta desafios [CRN03]. Dentre os desafios citados pelo autor, inclui-se a necessidade de criação de componentes reutilizáveis que possam ser largamente utilizados, podendo assim trazer benefícios econômicos para a empresa.

Nesse trabalho foi realizado um levantamento bibliográfico sobre processos e métricas para desenvolvimento baseado em componentes. Com base nesse estudo pode-se verificar a carência na literatura de modelos para análise de retorno de investimento para desenvolvimento de componentes, em particular modelos que combinassem fatores quantitativos e qualitativos.

Nesse contexto, o presente trabalho propõe um modelo de análise de viabilidade para criação de componentes reutilizáveis. O modelo proposto une fatores quantitativos e qualitativos para possibilitar a geração de cenários riscos sobre o retorno de investimento no desenvolvimento de componentes. Juntamente com o modelo foi desenvolvida uma ferramenta para auxiliar à aplicação do modelo.

8.1 TRABALHOS RELACIONADOS FRENTE AO MODELO PROPOSTO

Para facilitar a comparação do modelo proposto frente às demais propostas, apresentadas nas seções 7.1 e 7.2, foi criada a Tabela 14, que tem como objetivo comparar um conjunto de características frente às três abordagens.

Se compararmos o modelo proposto ao apresentado em [POU93], iremos verificar que o de [POU93] não incorpora quaisquer questões referentes a riscos no seu conjunto de métricas. Outra deficiência desse modelo é a subjetividade de algumas de suas métricas (como as equações 10 e 14 apresentadas na Seção 3.2) e o uso exclusivo de número de linhas de código como unidade de medida de software.

Se comparado à proposta de [MAL93], o modelo proposto evolui o conceito de risco, propondo um conjunto de riscos que podem dificultar o reuso do componente. Outro problema que dificulta o uso do modelo proposto em [MAL93] é incompletude da proposta, faltando um subconjunto de métricas que auxiliem o cálculo dos custos de desenvolvimento e reuso do componente.

Tabela 14. Comparativo das propostas de modelos

Proposta	[POU93]	[MAL93]	Modelo Proposto
Incorporação de fatores quantitativos	Sim	Sim	Sim
Incorporação de riscos ao reuso	Não	Sim	Sim
Especificação dos fatores qualitativos	Não	Não	Sim
Modelo completo ²	Sim	Não	Sim
Ferramenta de apoio à aplicação do modelo	Não	Não	Sim
Técnica de cálculo de ROI	VPL	VPL	VPL

8.2 CONTRIBUIÇÕES

O modelo de análise de viabilidade de componentização de software é a contribuição do presente trabalho. O modelo proposto, justifica-se por:

- Ter sua especificação elaborada com base em um levantamento bibliográfico sobre métricas para DBC e métodos de ROI;
- Definir um conjunto de entradas quantitativas e qualitativas que visam estabelecer diferentes cenários de reuso;
- Servir para auxiliar engenheiros de software na tomada de decisão sobre a criação de componentes de software.

² Apresenta todas as métricas necessárias para sua aplicação.

8.3 TRABALHOS FUTUROS

A partir do trabalho apresentado neste documento é possível apontar novos trabalhos, que servirão para dar continuidade na pesquisa:

- Avaliação e refinamento do modelo proposto através da aplicação do mesmo em diversos projetos e sobre diferentes componentes em um ambiente de desenvolvimento de software real;
- Integrar ao modelo métricas que auxiliem também o processo de tomada de decisão sobre a incorporação de componentes a uma aplicação, cobrindo assim mais amplamente as métricas de processo;
- Integrar o modelo de análise de viabilidade de componentização de software ao processo de desenvolvimento baseado em componentes. Possibilitando identificar o momento de tomada de decisão e quais os artefatos devem ser gerados dentro do processo para alimentar o modelo.

REFERÊNCIAS BIBLIOGRÁFICAS

- [ABT00] ABTS, C.; BOEHM, B.; CLARK, E. “COCOTS: A COTS Software Integration Cost Model – Model Overview and Preliminary data Findings”. In: Proceedings of the ESCOM-SCOPE 2000 Conference, April 2000, Munich, Germany, Shaker Publ., pp. 325-333.
- [BAC00] BACHMAN, F.; BASS, L.; BUHMAN, C.; COMELLA-DORDA, S.; LONG, F.; ROBERT, J.; SEACORD, R.; WALLNAU, K. “Volume II: Technical Concepts of Component-Based Software Engineering”. Capturado em: <http://www.dimap.ufrn.br/~jair/ES/artigos/cbse2.pdf>, Junho 2003.
- [BAR87] BARNES B.; DUREK, T.; GAFFNEY, J.; PYSTER, A. “A Framework and Economic Foundation for Software Reuse”. In: Proceedings of the Workshop on Software Reusability and Maintainability, Oct. 1987, pp. 77-88.
- [BAR91] BARNES, B.; BOLLINGER, T. “Making Reuse Cost-Effective”. IEEE Software, vol. 8, Jan. 1991, pp. 13-24.
- [BER00] BERGNER, K.; RAUSCH, A.; SIHLING, M.; VILBIG, A. “On Putting the Parts Together – Concepts, Description Techniques, and Development Process for Componentware”. In: Proceedings of the 33rd Hawaii International Conference On System Sciences (Hicss 00), 2000, Hilton Waikoloa Village, HA. Capturado em: <http://csdl.computer.org/comp/proceedings/hicss/2000/0493/08/04938050.pdf>, Julho 2004.
- [BOE81] BOEHM, B. “Software Engineering Economics”. Englewood Cliffs, NJ : Prentice Hall, NJ, 1981.
- [BOE88] BOEHM, B. “A spiral model of software development and enhancement”. Computer, vol. 21, n.5, Mai. 1988, pp. 61-72.
- [BOE95] BOEHM, B.; CLARK, B.; HOROWITZ, E.; WESTLAND, C.; MADACHY, R.; SELBY, R. “Cost Models for Future Software Lifecycle Processes: COCOMO 2.0”. In: Annals of Software Engineering, Vol. 1, Number 1, Dec. 1995, pp. 57-94.

- [BOE97] BOEHM, B. "COCOMO II Model Definition Manual". Los Angeles, CA: Science Department, University of Southern California, 1997.
- [BOX04] BOXALL, M.; ARABAN, S. "Interface Metrics for Reusability Analysis of Components". In: Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'2004), 2004, Australia, Melbourne.
- [BRO00] BROWN, A. W. "Large-scale component-based development". Upper Saddle River: Prentice Hall, 2000. 286p.
- [CHO01] CHO, E.; KIM, M.; KIM S. "Component Metrics to Measure Component Quality". In: Eighth Asia-Pacific Software Engineering Conference 2001 (Dec 4-7, 2001), Macau, China, pp. 419-426.
- [CON04] CONSTRUCTIVE COTS. Desenvolvido pela USC. Capturado em: <http://sunset.usc.edu/research/COCOTS/>, Outubro 2004.
- [CRN02] CRNKOVIC, I.; LARSSON, M. "Building reliable component-based software systems". Norwood, MA: Artech House, 2002. 413p.
- [CRN03] CRNKOVIC, I. "Component-Based Software Engineering – New Challenges in Software Development". In: Proceedings of the 25th International Conference on Information Technology Interfaces (ITI'03), (Jun 16-19, 2003), Cavtat, Croatia. pp. 9-18.
- [CRU03] CRUZ, C.; RIBEIRO, U. "Metodologia Científica: Teoria e Prática". Rio de Janeiro: Axcel Books, 2003, 205p.
- [DAG03] DAGNINO, A.; SRIKANTH, H.; NAEDELE, M.; BRANTLY, D. "A Model to Evaluate the Economic Benefits of Software Components Development". In: IEEE International Conference on Systems, Man, and Cybernetics, (Oct. 5-8, 2003), Vol. 4, Washington, D.C., USA, pp. 3792-3797.
- [EZR02] EZRIN, M.; MORISIO, M.; TULLY, C. "Practical Software Reuse", London: Springer-Verlag, 2002. 222p.
- [FAV96] FAVARO, J. "A Comparison of Approaches to Reuse Investment Analysis". In: Proceedings of the 4th International Conference on Software Reusability, Orlando, Apr. 1996.

- [FIC01] FICHMAN, R. G.; KEMERER, C. F. "Incentive Compatibility and Systematic Software Reuse". The Journal of Systems and Software, Vol. 57, Issue 1, 27 april 2001, pp. 45-60.
- [FRA96a] FRAKES, W.B.; FOX, C.J. "Quality Improvement Using a Software Reuse Failure Modes Model". In: IEEE Transactions on Software Engineering, Vol. 22, Issue 4, Apr. 1996, pp.274-279.
- [FRA96b] FRAKES, W.; TERRY, C. "Software Reuse: Metrics and Models". ACM Computing Surveys (CSUR), Vol. 28, Issue 2. Jun.1996, pp. 415-435.
- [GAF92] GAFFNEY, J.; CRUICKSHANK, R. "A General Economics Model Of Software Reuse". In: Proceedings of the 14th International Conference on Software Engineering, 1992, Australia, Melbourne, pp. 327-337.
- [GRA92] GRADY, R. "Practical Software Metrics for Project Management and Process Improvement". Englewood, USA: PRENTICE HALL, 1992, 270p.
- [HOO91] HOOPER, J.; CHESTER, R. "Software Reuse: Guidelines and Methods". New York: Plenum Press, 1991, 196p .
- [KAN87] KANG, K. "A Reuse-Based Software Development Methodology". In Proceedings of the Workshop On Software Reuse. Boulder, CO, October, 1987.
- [LIM94] LIM, W. "Effects of Reuse on Quality, Productivity, and Economics". IEEE Software, sep. 1994, Vol. 11, Issue 5, pp. 23-30.
- [LIM98] LIM, W. C. "Managing Software Reuse". NJ, USA: Prentice Hall, 1998. 480 p.
- [MAL93] MALAN, R.A.; WENTZEL, K. "Economics of Reuse Revisited". Technical Report HPL 93-31., Palo Alto, CA: Hewlett-Packard Laboratories, Apr. 1993.
- [MAR92] MARGANO, J.; LINDSEY, L. "Software Reuse in the Air Traffic Control Advanced Automation System". Joint Symposia And Workshops: Improving The Software Process And The Competitive Position. Alexandria, VA (Apr. 29- May 3, 1992).

- [OLI82] OLIVEIRA, J. “Engenharia Econômica: uma abordagem às decisões de investimento”. São Paulo: McGraw-Hill do Brasil, 1982, 173p.
- [PFL96] PFLEEGER, S. L. “Measuring Reuse: A Cautionary Tale”. IEEE Software, Jul. 1996, vol. 13, Issue 34, pp. 118-127.
- [POU93] POULIN, J.S.; CARUSO, J.M. “A reuse metrics and return on investment model”. In: Software Reusability. 1993, Itália, Lucca, pp.152-167.
- [POU93a] POULIN, J.S.; CARUSO, J.M.; HANCOCK, D. R. “The Business Case for Software Reuse”. IBM Systems Journal. vol. 32, Out 1993, pp. 567-594.
- [POU94] POULIN, J.S. “Measuring Software Reusability”. In: Proceedings of Third International Conference on Software Reuse: Advances in Software Reusability, Rio de Janeiro, Brasil (Nov. 1-4, 1994), pp. 126-138.
- [RAJ94] RAJLICH, V.; SILVA, J. “A Case of Study of Software Reuse in Vertical Domain”. In: Proceedings of 4th System Reengineering Workshop, Monterey, CA (Feb 8-10, 1994). John Hopkins University: Applied Physics Laboratory, APL Research Ctr. Report RM 94-003, pp. 67-76.
- [REI97] REIFER, D. “Practical Software Reuse”. Nova York: John Wiley & Sons, Inc, 1997. 374 p.
- [SAM97] SAMETINGER, J. “Software Engineering with Reusable Components”. New York: Springer, 1997, 271p.
- [SED01] SEDIGH-ALI, S.; GHAFOOR, A.; PAUL, R.A. “Software engineering metrics for COTS-based systems”. IEEE Computer, vol. 34, mai. 2001, pp. 44-50.
- [SPC92] SOFTWARE PRODUCTIVITY CONSORTIUM, “Reuse Adoption Guidebook”, Versão 01.00.03, Report SPC-92051-CMC, Novembro de 1992.
- [SPC93] SOFTWARE PRODUCTIVITY CONSORTIUM. “Reuse Adoption Guidebook”, Versão 02.00.05, Report SPC-92051-CMC, Novembro de 1993.

- [SZY99] SZYPERSKI, C. “Component Software: beyond object-oriented programming”. Harlow: Addison-Wesley, 1999, 411p.
- [UML03] UML. “Unified Modeling Language Specification”. Capturado em: <http://www.uml.org>, Maio 2003.
- [VIR03] VIRTANEN, P. “Measuring and Improving Component-Based Software Development”. Tese de doutorado. Finlândia: University of Turku: Faculty of Mathematics and Natural Sciences, 2003. 204 f.
- [WAS03] WASHIZAKI, H.; YAMAMOTO, H.; FUKAZAWA, Y. “A Metrics Suite for Measuring Reusability of Software Components”. In Proceedings of the 9TH International Software Metrics Symposium, (Sep. 3-5, 2003) Australia, Sydney, pp. 211-223.

**APÊNDICE I - INSTRUMENTO DE COLETA DE DADOS DA
PESQUISA DE CAMPO**

I. Formulário de Caracterização

CARACTERIZAÇÃO DO DESENVOLVEDOR					
Nome _____					
Formação Acadêmica:					
<input type="checkbox"/> Doutorado	<input type="checkbox"/> Mestrado	<input type="checkbox"/> Especialização	<input type="checkbox"/> Graduação		
<input type="checkbox"/> Técnico	<input type="checkbox"/> Outra: _____				
Formação Geral:					
Qual é sua experiência anterior com desenvolvimento baseado em componentes na prática ? (marque aqueles itens que melhor se aplicam)					
() nunca desenvolvi software baseado em reutilização.					
() tenho desenvolvido software com foco em artefatos reutilizáveis para uso próprio.					
() tenho desenvolvido software com foco em artefatos reutilizáveis como parte de uma equipe, relacionado a um curso.					
() tenho desenvolvido software com foco em artefatos reutilizáveis como parte de uma equipe, na indústria.					
Por favor, explique sua resposta. Inclua o número de semestres ou número de anos de experiência relevante em desenvolvimento. (E.g. "Eu trabalhei por 10 anos como programador na indústria")					

Você já utilizou ou utiliza alguma técnica para apoio ao desenvolvimento de componentes reutilizáveis?					
() Sim () Não Indique a(s) técnica(s)?					

POR FAVOR, INDIQUE O GRAU DE SUA EXPERIÊNCIA NESTA SEÇÃO SEGUINDO A ESCALA DE 5 PONTOS ABAIXO:					
1 = nenhum	2 = estudei em aula ou em livro				
3 = pratiquei em 1 projeto em sala de aula	4 = usei em 1 projeto na indústria				
5 = usei em vários projetos na indústria					
Experiência em projeto de sistemas	1	2	3	4	5
Domínio de Conceitos de reutilização de software	1	2	3	4	5
Experiência com métricas de software	1	2	3	4	5

II. Formulário de Avaliação das Métricas de Análise de Viabilidade de Componentização de Software

O propósito único deste questionário é apoiar o pesquisador no levantamento de dados para melhor entendimento do estudo experimental em questão. Não será utilizado para avaliar o participante.

a. Avaliação Geral do Modelo de Análise de Viabilidade

1) O modelo de análise de viabilidade de componentes auxiliou na decisão de componentização de software? Justifique sua resposta caso a opção escolhida seja “não” ou “parcialmente”:	<input type="checkbox"/> Sim <input type="checkbox"/> Não <input type="checkbox"/> Parcialmente

2) Você julga que a possibilidade de geração de diferentes cenários através da parametrização das taxas de adaptabilidade ajuda na análise de viabilidade para criação de componentes? Justifique sua resposta caso a opção escolhida seja “não” ou “parcialmente”:	<input type="checkbox"/> Sim <input type="checkbox"/> Não <input type="checkbox"/> Parcialmente

3) Você julga que a possibilidade de geração de diferentes cenários através da distribuição de custos em diferentes projetos ajuda na análise de viabilidade para criação de componentes? Justifique sua resposta caso a opção escolhida seja “não” ou “parcialmente”:	<input type="checkbox"/> Sim <input type="checkbox"/> Não <input type="checkbox"/> Parcialmente

b. Avaliação das Entradas Quantitativas

4) Você julga que as variáveis de entrada do modelo são suficientes para estipular custos no desenvolvimento e reuso de componentes? Justifique sua resposta caso a opção escolhida seja “não” ou “parcialmente”:	<input type="checkbox"/> Sim <input type="checkbox"/> Não <input type="checkbox"/> Parcialmente

c. Avaliação das Entradas Qualitativas

5) Você julga que as variáveis qualitativas do modelo (Manutenção, Documentação, Performance, Obsolescência Funcional e Obsolescência Técnica) são suficientes para avaliação do ciclo de vida do componente? Justifique sua resposta caso a opção escolhida seja “não” ou “parcialmente”:	<input type="checkbox"/> Sim <input type="checkbox"/> Não <input type="checkbox"/> Parcialmente

APÊNDICE II – RELAÇÃO DAS EQUAÇÕES APRESENTADAS

Número	Equação
1	$ER_{(m < 25\%)} = TC \times ERLM$
2	$ER_{(m > 25\%)} = TC \times ERMM$
3	$ETRM = ER_{(m < 25\%)} + ER_{(m > 25\%)}$
4	$ETRSM = TC \times ERNM$
5	$ETR = ETRM + ETRSM$
6	$ETER = TC \times ERE$
7	$ETDBC = TECN + ETR + ETER$
8	$ETT = TP$
9	$ET_{salvo} = ETT - ETDBC$
10	$CPPR = \sum_{i=1}^j b_i = \sum_{i=1}^k c_i$
11	$CPD = CR \times 0.8 \times CCN$
12	$CPM = CR \times TE \times CCN$
13	$CPPR = CPD + CPM$
14	$CAD = \sum_{i=1}^j c_i - \sum_{i=1}^k b_i$
15	$CAD = (CRER - 1) \times QCC \times CDCN$
16	$N_0 = \frac{E}{1 - B}$
17	$VPL = \sum_{t=1}^n \frac{FC_t}{(1+k)^t} - II$
18	$ROI = RCA + ORCA - ADC$
19	$ROI = \left[\sum_{i=1}^n (CNR_i - CRC_i) \times PR_i \right] - CDC + CER$
20	$CPR = \left[\sum_{i=1}^{NRA} (CNR - CRC) \right] - CDC$
21	$CI = TF \times EE \times CD$
22	$CRC = TF \times TAC \times CD$
23	$CNR = CRD + CRM$
24	$CD = TF \times CD$
25	$CM = TF \times TE \times CMC$
26	$CPR = \left[\sum_{i=1}^{NRA} \left(\frac{CNR - CRC}{(1 + TMA)^{ano}} \right) \right] - CDC$
27	$CPR = \left[\sum_{i=1}^{NRA} \left(\frac{CNR - CRC}{(1 + TMA)^{ano}} \right) \times PR \right] - CDC$