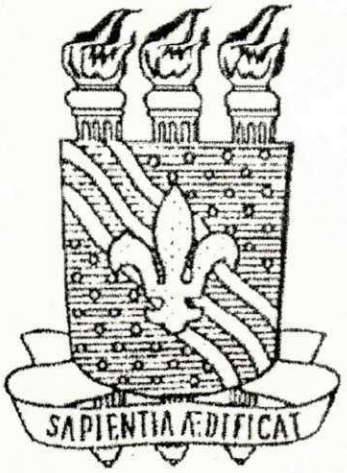


UNIVERSIDADE FEDERAL DA PARAÍBA

CENTRO DE CIÊNCIAS E TECNOLOGIA

DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO

COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Produção de *Software Add-On* – Sugestões de Procedimentos para o Ciclo de Vida do Produto

Maurício Floriano Galimberti

Campina Grande - PB

Setembro de 1997

UNIVERSIDADE FEDERAL DA PARAÍBA

CENTRO DE CIÊNCIAS E TECNOLOGIA

DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO

COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Maurício Floriano Galimberti

Produção de *Software Add-On* – Sugestões de Procedimentos para o Ciclo de Vida do Produto

Dissertação apresentada ao curso de MESTRADO EM INFORMÁTICA da Universidade Federal da Paraíba, em cumprimento às exigências para obtenção do Grau de Mestre.

Área de Concentração: Sistemas de Software

José Antão Beltrão Moura

(Orientador)

Campina Grande - PB

Setembro de 1997

G158p Galimberti, Maurício Floriano.
Produção de software Add-On : sugestões de procedimentos para o ciclo de vida do produto / Maurício Floriano Galimberti. - Campina Grande, 1997.
91 f.

Dissertação (Mestrado em Informática) - Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, 1997.
Referências.
"Orientação : Prof. Dr. José Antão Beltrão Moura".

1. Engenharia de Software. 2. Software - Sistemas. 3. Software Add-On - Produto - Vida Útil. 4. Dissertação - Informática. I. Moura, José Antão Beltrão. II. Universidade Federal da Paraíba - Campina Grande (PB). III. Título

CDU 004.41(043)

PRODUÇÃO DE SOFTWARE ADD-ON-SUGESTÕES DE PROCEDIMENTOS PARA O CICLO DE VIDA DO PRODUTO

MAURICIO FLORIANO GALIMBERTI

DISSERTAÇÃO APROVADA EM 01.09.1997

PROF. JOSÉ ANTÃO BELTRÃO MOURA, Ph.D
Presidente

PROF. MARCELO ALVES DE BARROS, Dr.
Examinador

PROF. FRANCISCO VILAR BRASILEIRO, Ph.D
Examinador

PROF. LUIZ MAURICIO FRAGA MARTINS, M.Sc

CAMPINA GRANDE - PB

"CH'IEN - O CRIATIVO promove sublime sucesso,
favorecendo através da perseverança."

I CHING

Agradecimentos

A meus Pais e minha Irmã que, mesmo com a distância e separação, sempre me incentivaram, depositando total segurança para que eu perseguisse meus sonhos e ideais.

A José Bussmann pelo companheirismo, confiança e indicação para trabalho em conjunto e a Luiz Maurício pela receptividade, amizade, e pelas oportunidades proporcionadas, além da experiência acadêmica aliada à proximidade com o mercado.

A meu orientador, pela sintonia que foi necessária durante o trabalho e transmissão de conhecimentos e práticas da produção de *software*.

A Nivaldo, pelas experiências e práticas técnicas a mim transmitidas.

A todos meus amigos e, especialmente, a Marcão, Jaque e Paty, que sempre estavam ali para levantar o moral e ao meu camarada DV, pela parceria que nos proporcionou horas translouquecidas de trilhas de moto pela natureza do interior de Campina Grande, fundamental para suportar as semanas de muito trabalho.

A todos colegas, professores e funcionários que colaboraram para a conclusão desta dissertação.

Aos profissionais da empresa Fácil Informática, pela recepção em Blumenau, experiência proporcionada e incentivo à continuidade das pesquisas.

Ao CNPq e CAPES, pelo suporte financeiro que tornou possível a elaboração deste trabalho.

Especial a meu avô Orlando, pela paz, tranquilidade e perseverança a mim transmitidas.

Resumo

O mercado especializado sugere a concentração de produtores de *software* em nichos específicos, visando aumentar ou até mesmo inovar recursos de um outro *software* existente no mercado.

O presente trabalho organiza o ciclo de produção deste tipo de *software* agregado (*software add-on*), antecipando dificuldades e sugerindo diretrizes a procedimentos ideais ao longo do processo, desde a concepção do produto, passando pelo desenvolvimento, preparação e concluindo com a disponibilização do mesmo ao mercado. O trabalho visa proporcionar um processo sistematizado e que otimize esforços e as chances de sucesso durante a produção e disponibilização de *software* do tipo *add-on*.

Palavras-Chave: *software add-on*, ciclo de produção de *software*, engenharia de *software*.

sistemas de software.

Abstract

The life cycle of "add-on" software (that is, software that enhances or extends the functionality of another) is examined in order to identify difficulties and to suggest procedures to either avoid or overcome them. Emphasis is given to marketing and commercial issues so that the chance of success of an add-on product in the market place is improved.

Key Words: add-on software, life cycle of software, software engineering.

Sumário

Lista de Figuras

Lista de Gráficos

Lista de Tabelas

Resumo Executivo

O segmento de *software add-on*, que agrega funcionalidades e/ou sugere um novo paradigma para outro produto de *software*, aparenta ser um bom canal para novos empreendimentos, podendo minimizar custos e riscos dentro do ciclo de vida do produto. O presente trabalho trata as atividades do processo de produção de *add-on*. Este resumo executivo forma um guia específico de "passos" a serem seguidos para se produzir *software add-on*, detalhando-se cada uma das atividades no decorrer da dissertação.

Palavras-chave: *software add-on*, processo de produção de *software*, ciclo de vida de *software*, *software* hospedeiro.

1 - Apresentação

Atualmente verifica-se que a indústria de *software* requer altos volumes de investimentos até a oferta do produto ao mercado. Esta realidade motiva a procura por novos mercados, mais lucrativos e com menores investimentos e riscos. Assim, acredita-se que o segmento de *software add-on* configura-se em uma boa alternativa para abarcar mercados ainda pouco explorados.

Consideram-se como produtos de *software* do tipo *add-on* aqueles que tenham como meta agregar/adicionar valor funcional a outros produtos de *software* hospedeiros. Estes, por sua vez, tornam-se mais úteis ou mais fáceis de serem utilizados.

Exemplos de *software add-on*, com seus respectivos hospedeiros (<add-on> ➔ <hospedeiro>), são:

- corretores ortográficos:

  CorrectSpell[1] ➔ processador de textos Word[2] ;

- "hifenizadores" ortográficos:

  International Hyphenator[1] ➔ processador de textos Word;

- ferramentas para auxílio e gerência de impressão de relatórios;
- filtros gráficos (largamente ofertados na Internet);
- localizadores de documentos:

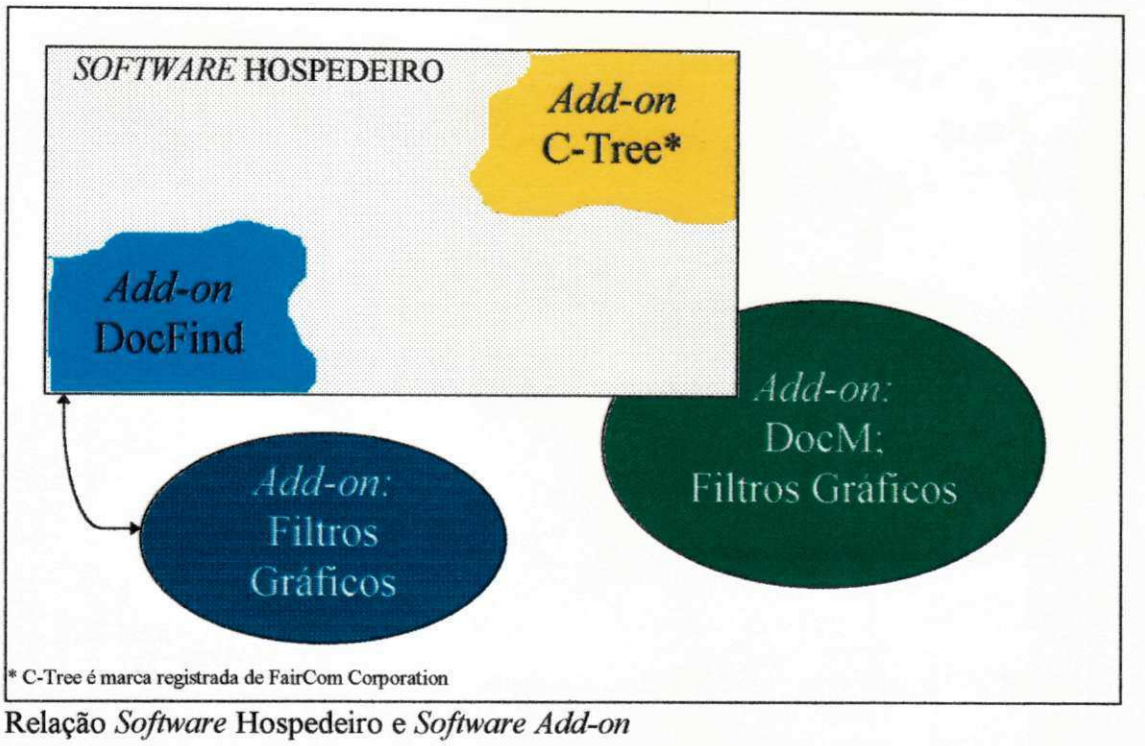
  DocFind[3] ➔ processador de textos Fácil

  DocM[3] ➔ processador de textos Word

[1] CorrectSpell e International Hyphenator são marcas registradas de INSO Corporation

[2] Word é marca registrada de Microsoft Corporation

[3] *Add-on* DocM e DocFind são de propriedade de Green Software Ltda.



Relação *Software* Hospedeiro e *Software Add-on*

O presente trabalho busca contribuir como um guia prático, parametrizando as principais atividades constantes no ciclo de produção de *software add-on*. Para melhor ilustrar o processo de produção do *add-on*, são utilizados dois estudos-de-casos, citados acima como os produtos *add-on* DocM e DocFind, projetados respectivamente para os processadores hospedeiros Word e Fácil, estando diferenciados na figura anterior devido a particularidades de integração e comercialização em relação ao *software* hospedeiro.

O modelo aqui definido segue o molde R-Cycle para o processo PDE (Produção, Disponibilização e Evolução) de *software* [MART93] e [MOUR94]. Este molde divide o processo em quatro fases distintas, sendo: concepção, desenvolvimento, preparação e disponibilização de *software*, onde em cada uma delas existem atividades específicas, inerentes a cada fase, e outras atividades que acompanham, com diferente intensidade, todo o ciclo de vida do *software*. O molde R-Cycle busca a flexibilidade, permitindo ser instanciado e calibrado de acordo com as necessidades de cada produto, e do nicho de

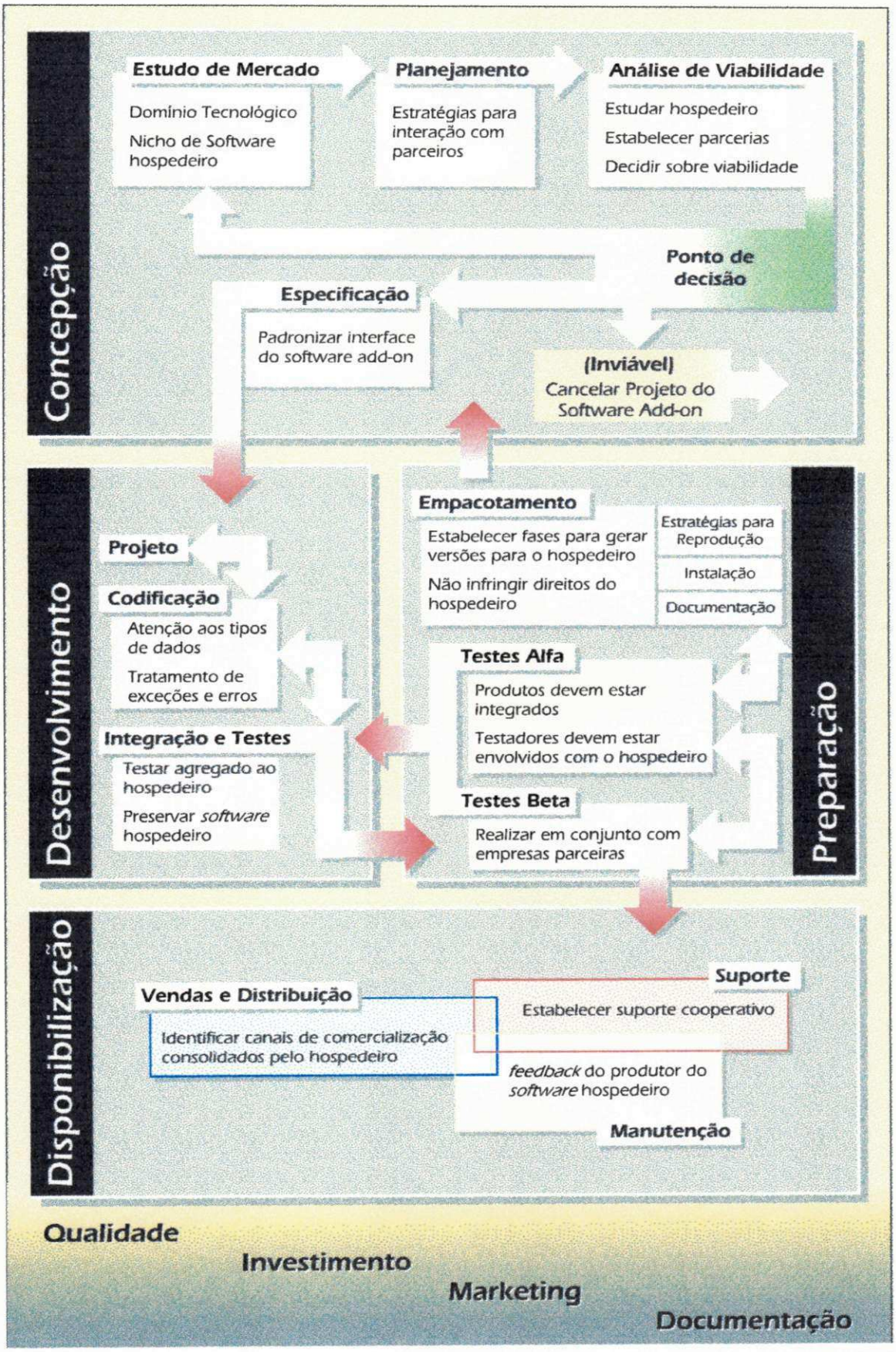
mercado por este visado. Por isto, nós aqui o adotamos para nortear nossos comentários e sugestões para a produção de *software add-on*.

2 - Produzindo *Software Add-on*

Software do tipo *add-on* tem como requisito principal a existência de outro *software* que o agregue, ou seja, um *software* hospedeiro. Com isso, o processo que envolve a produção de um *add-on* é acompanhado, quase que completamente, pela interação entre os produtores do *software add-on* e hospedeiro.

No entanto, a maior dificuldade está em se identificar o nicho de *software* a ser atacado e garantir que aquele escolhido realmente seja viável tanto técnica quanto comercialmente. Isto torna a fase de concepção a mais importante dentro do processo de produção do *add-on*.

A partir da Figura do modelo para *software add-on*, aborda-se a seguir o seu processo de produção, seguindo a estrutura do molde R-Cycle e identificando-se as nuances relativas à produção de *software* do tipo *add-on*.



Instanciação do Molde R-Cycle para o Processo PDE de *Software Add-on*

2.1 - Concepção

A fase de concepção é a fase inicial do processo de produção e marca o ponto onde é idealizado o *software* a ser produzido, estabelecendo-se as estratégias para a execução do projeto.

Esta fase é disparada pela descoberta de oportunidades de mercado, e, especificamente no caso de *software add-on*, tais oportunidades devem apontar para a evolução de um *software* já disponível no mercado, o que torna altamente importante a adoção dos testes de usabilidade já no início da fase de concepção.

Sendo assim, a instanciação desta fase requer necessariamente como atividade inicial, que sejam analisados os segmentos de mercado, procurando o nicho de mercado a ser almejado. Em seguida, deve-se planejar o estudo de viabilidade e se for o caso, executá-lo.

A análise de viabilidade - considerando-se o relacionamento técnico, de *marketing* e comercial com o produtor hospedeiro - se concretiza no ponto chave de decisão de todo o processo para *software add-on*.

Levantamento dos Segmentos de Mercado

- Partir da dimensão do empreendimento do produtor do *software add-on*;
- A partir da tecnologia dominante, busca-se qual mercado se deseja levantar;
- Identificar oportunidades funcionais no mercado segmentado;
- Questionar o público alvo: algumas questões a este respeito, utilizadas com o público no estudo-de-caso DocM-Word, foram:
  - Qual(is) editor(es) de textos utilizado(s)?
  - Aproximadamente qual o tempo gasto para localizar um arquivo antigo?
  - Como faz para localizá-lo?

- Enfim, seria produtivo localizar os documentos pesquisando através dos conteúdos dos documentos e não apenas pelo nome do arquivo?

- Constituir o nicho de *software* a ser atacado;

- Levantar requisitos;

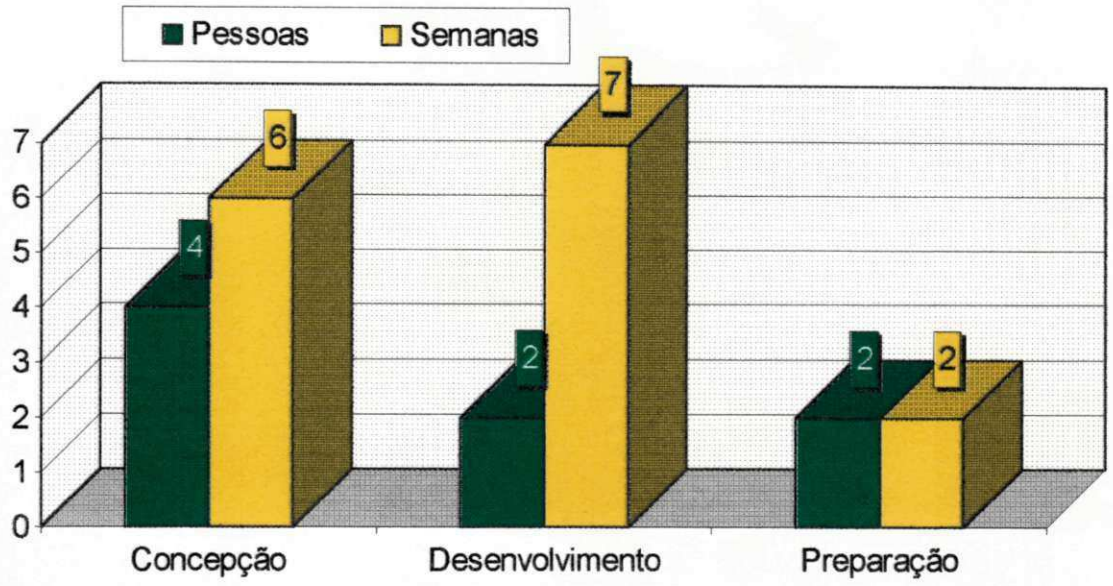
Planejamento

- Organizar estudos/análise de viabilidade técnica e comercial e criar as respectivas equipes;

- Definir estratégias para interação com parceiros e equipes responsáveis;

- Estimar custos e prazos;

Considera-se ainda que, no processo de produção de *software add-on*, o controle de qualidade deve ser aplicado mais intensamente nas fases iniciais de concepção e desenvolvimento. O Gráfico a seguir (apesar de não cobrir o ciclo por completo) identifica estas duas fases, para o *add-on* DocFind, como sendo as mais prolongadas e que necessitam maiores esforços e investimentos, principalmente pelos aspectos de viabilidade e integração com parceiros.



* Não há instanciação para a fase de Disponibilização

Projeto DocFind: Relação Tempo-Pessoa para Fases do Ciclo de Produção de *Add-on*

Assim, o Gráfico identifica, para execução de cada fase em uma semana, a necessidade de 24 pessoas na concepção e 14 no desenvolvimento, contrastando com os 4 indivíduos necessários à preparação do *add-on*. Salienta-se que esta relação somente é válida para expressar os esforços necessários para se desempenhar cada fase.

Análise de Viabilidade: ponto de decisão do processo.

A análise de viabilidade é vital para a produção do *add-on*. Assim, aqui é o momento de se decidir sobre a continuidade ou não do projeto para o *software add-on*.

- Estudar minuciosamente o *software* hospedeiro, buscando identificar aspectos técnicos favoráveis à integração;
- Descrever vantagens e desvantagens para cada solução vislumbrada;
- Identificar perspectiva das futuras *releases* do *software* hospedeiro;

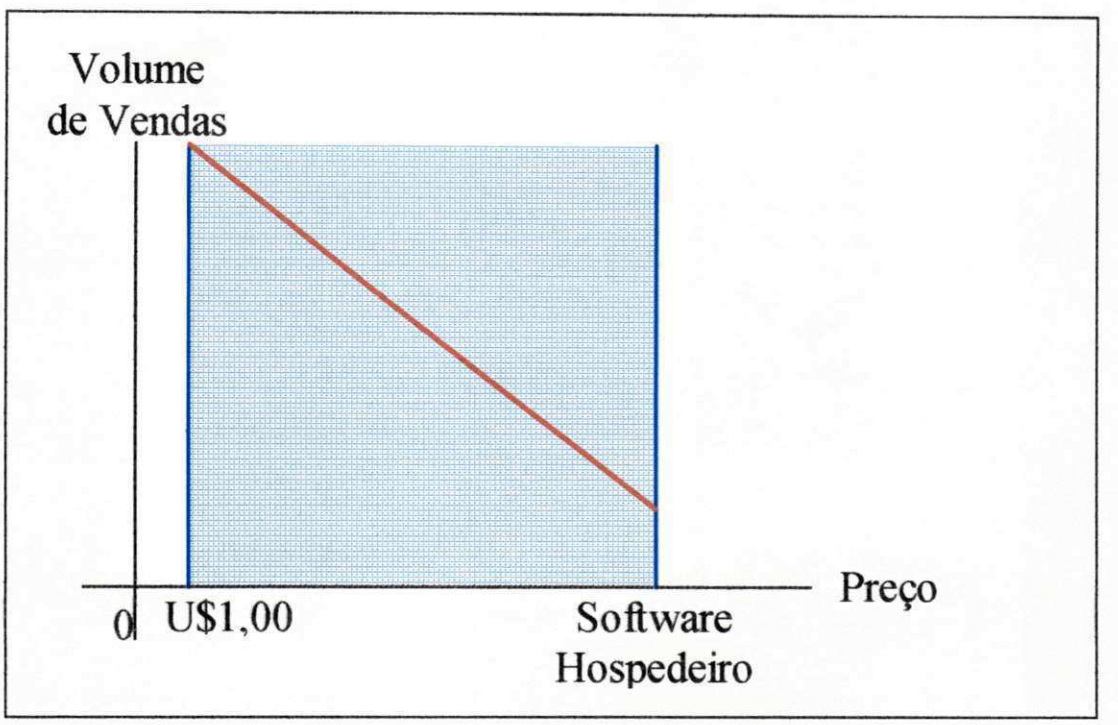
- Levantar alternativas para disponibilização integrada dos produtos de *software*;
- Buscar estabelecer parceria comercial;
- Analisar riscos;
- Após estas tarefas, decidir sobre continuidade ou não da produção do *software add-on*;

Interação com o Parceiro

- Quando da identificação de oportunidades, ao se adquirir a cópia do *software* hospedeiro;
- Quando da análise de viabilidade comercial;
- Propor alternativas de integração entre os produtos;
- Expor dúvidas, buscando soluções e melhores estratégias de ligação entre os produtos de *software*;
- Identificar as plataformas de desenvolvimento usadas pelos dois produtos de *software;*
- Estabelecer a interface de comunicação entre os produtos de *software;*
- Definir os momentos em que as funcionalidades do *add-on* serão executadas pela aplicação que o recebe;
- Identificar o tipo de interface de usuário solicitada;
- Equipe de marketing estabelece contato com relação aos aspectos mercadológicos e comerciais;
- Considerar, junto à empresa parceira, os aspectos legais inerentes a cada alternativa de comercialização existente;

Com relação aos rendimentos, vale estabelecer um intervalo possível para o preço a ser atribuído *ao add-on*. Partindo-se do foco de marketing, deve-se considerar a realidade do produtor hospedeiro, conhecendo a base

instalada, o volume e a média de vendas e o preço final do *software* hospedeiro.



Intervalo para Estabelecimento de Preço do *Software Add-on*

Além disso, analisando-se os benefícios a serem trazidos pelo *software add-on*, que podem ser quantificados inclusive pelos testes de usabilidade, e a sua importância em relação ao hospedeiro, de forma geral a linha vermelha poderá servir como ferramenta de decisão de preço para o *add-on*. Identifica-se assim uma relação inversamente proporcional entre preço do *add-on* e volume de vendas, onde parte-se do preço de U$1,00 para um volume de vendas "elevado" (isto é relativo aos custos envolvidos para a produção do *add-on*), até um preço do *add-on* se aproximando do preço de venda do *software* hospedeiro quando da comercialização dos produtos para um nicho mais específico e restrito. Para tanto, deve-se conhecer os custos envolvidos na produção do *add-on* e qual o montante requerido para que se comece a contabilizar lucros.

Especificação

- Partir da análise de requisitos;

- Definir projeto arquitetural: objetivos do *software add-on*;

- Especificar funcionalidades do produto;

- Estabelecer padronização da interface;

- Criar/prototipar a interface do *software add-on*;

Conforme a interface a ser adotada com o usuário, o modelo para *add-on* sugere a adoção de um conjunto de *check-lists*. A seguir são listados estes *check-lists*, salientando-se a importância das técnicas de interação adotadas pelo hospedeiro, as cores presentes em sua interface, os métodos de ajuda utilizados e a maneira que o usuário final possui para interagir com o produto hospedeiro. Maiores detalhes, sobre como montar o conjunto de *check-lists*, encontram-se no capítulo 2 da dissertação.

1 - Análise de Ações e Formas de Alcance

2 - Análise de Técnicas para Interação:

ESTILO / TÉCNICA
Seleção de Menus
Preenchimento de Formulários
Linguagem Natural
Manipulação Direta
Janelas de Tarefas
Ícones p/ Arquivos e Diretórios
Caixas de Diálogo
Botões
Sliders
Check Boxes
Listas

3 - Linguagem de Comandos

4 - Métodos de Destaque

5 - Uso de Cores:

		CORES PRINCIPAIS DA TELA	CORES PISCANTES	
			Principal	Destaque
	Cinza Claro			
	Cinza Escuro			
	Amarelo Escuro			
	Vermelho Escuro			
	Magenta Escuro			
	Verde Escuro			
	Ciano Escuro			
	Azul Escuro			
	Branco			
	Amarelo			
	Vermelho			
	Magenta			
	Verde			
	Ciano			
	Azul			
	Preto			

6 - Métodos de Ajuda:

MÉTODOS DE AJUDA
Manuais On-line
Manuais Impressos
Ajuda Sensível ao Contexto
Tutoriais, Demonstrações e Animações On-line

7 - Dispositivos de Interação

ENTRADA/SAÍDA	DISPOSITIVOS DE INTERAÇÃO
DISPOSITIVOS DE ENTRADA	
	Teclado/Teclas Especiais
	Posicionadores/Selecionadores: Mouse Trackerball Joystick Caneta ótica Teclas de controle Tela sensível ao toque
	Leitoras de código: Barras Magnético
	Tablete digitalizador
	Scanner
	Luvas/Capacetes/Macacões
DISPOSITIVOS DE SAÍDA	
	Monitores de vídeo
	Áudio
	Impressoras

⇨ ***Check-list* Final**: Requisitos para a Interface do *Add-on*

Os *check-lists* tem como objetivo padronizar a interface do *software add-on* de forma a sintonizá-la com a do hospedeiro. Este foi o caso da interface do *add-on* DocM, onde os padrões adotados coincidem com os verificados no processador Word. Tal padronização é efetuada e analisada a partir da prototipação da interface do *software add-on*, ou obtendo-se informações com o produtor do *software* hospedeiro.

2.2 - Desenvolvimento

A fase anterior se caracterizou por aspectos de viabilidade e planejamento, afunilando-se através da atividade de especificação, que permite a entrada à fase de desenvolvimento já na especificação de mais baixo nível, ou

definição lógica do projeto do *software add-on*. No entanto, isto é uma característica do molde R-Cycle e, no caso de *software add-on*, não existem muitas diferenças, para a fase de desenvolvimento, dos outros tipos de *software*.

Projeto

- Partir da especificação funcional, demarcando funções, operações, e objetos;
- Definir comportamento dos componentes da interface;
- Organizar documentação que permita elaborar os manuais do usuário;

Codificação

- Criar classes, sub-classes e atributos dos objetos antes definidos;
- Verificar cada operação definida para os objetos de forma a transformá-las em protótipos dos métodos públicos de cada classe de objeto;

Integração e Testes

- Integrar módulos do *software*;
- Agregar *software add-on* ao hospedeiro;
- Testar integração e agregação e garantir confiabilidade, não expondo ao risco os dados mantidos pela aplicação hospedeira;
- Preservar a integridade provida pela aplicação hospedeira;
- Marcas as primeiras reuniões técnicas com objetivos de demonstração e testes alfa e beta.

2.3 - Preparação

A fase de preparação deve transformar o *software*, considerado finalizado pelo desenvolvimento, em um produto destinado ao mercado,

aplicando a este um rigoroso controle de qualidade, garantindo a integridade do mesmo para que seja comercializado.

São desenvolvidos todos os procedimentos de empacotamento e testes, juntamente com a documentação de aceitação dos usuários envolvidos nas atividades de testes.

Empacotamento

- Criar procedimentos específicos para instalação, proteção ao *software* e reprodução (controle de qualidade);
- Identificar a demanda existente;
- Confeccionar manuais: documentação
- Sistematizar o processo de reprodução;
- Estabelecer as fases em que o produtor do *software* hospedeiro venha a requerer o *software add-on*;
- Gerar e repassar cópias do *software add-on*;

Testes Alfa (Testes Internos)

- *Software* hospedeiro e *add-on* já devem estar integrados;
- Estabelecer dois grupos de testadores: o primeiro com pessoas que dominam o *software* hospedeiro, buscando-se o diferencial trazido pelo *add-on*; o segundo grupo dará o *feedback* sobre a usabilidade dos produtos em questão;

Testes Beta (Testes Externos Junto aos Clientes)

- Primeira etapa: formada pelos usuários e técnicos das empresas parceiras do *add-on*. Busca-se analisar as informações advindas de cada empresa;
- Priorizar correção de erros e carências mais comumente apontadas pelas empresas parceiras;

- Segunda etapa: utiliza da base de usuários estabelecida para o *software* hospedeiro;
- Disponibilizar versão beta ao público usuário indicado por parceiros, a fim de viabilizar uma nova versão do(s) *software*(s) hospedeiro(s);

2.4 - Disponibilização

A disponibilização do *add-on* tem como meta suportar técnica e comercialmente o produto de *software* no mercado. Para isto, deve-se ativar procedimentos que se concentrem em estabelecer as melhores alternativas para vendas e distribuição do *software*, criando a estrutura necessária para o suporte técnico ao produto e futura manutenção.

Durante toda a fase de disponibilização, o produtor do *software add-on* deve buscar aproveitar-se da estrutura já existente do *software* hospedeiro. Assim, pode-se utilizar dos canais de distribuição já estabelecidos, bem como adequar a operação do suporte técnico com a estrutura de suporte montada para o *software* hospedeiro.

Vendas e Distribuição

- A partir do mercado alvo previamente estabelecido, utilizar das alternativas existentes para comercialização;
- Analisar a base instalada do *software* hospedeiro e utilizar de canais de comercialização estabelecidos;
- Estabelecer acordos e estratégias para disponibilização e divulgação conjuntas dos produtos de *software*;
- Estabelecer estratégias para mídias alternativas (Internet);

Suporte

- Buscar prover suporte em conjunto com o produtor do *software* hospedeiro;

- De acordo com a alternativa de suporte adotada, deve-se criar a central de suporte;
- Buscar soluções para operação do suporte via *Internet*;

Manutenção

- Ativar gerência e marketing para buscar *feedback* dos usuários;
- Obter *feedback* do suporte, através também da gerência e marketing;
- Obter *feedback* das empresas parceiras: evolução do *add-on* em função do *software* hospedeiro;
- Priorizar mudanças/alterações no produto;
- Efetuar alterações e ativar geração de novas versões.

3 - Conclusão

Acredita-se que o entendimento das diretrizes expostas para se produzir *software add-on* venha a proporcionar um processo mais sistematizado para a produção deste tipo de *software*, otimizando esforços e as chances de sucesso.

Aqui, não se pretendeu tornar estático o processo de produção de *software add-on* mas, sim, apontar para caminhos menos tortuosos. Assim, o empreendedor deve encarar cada medida como sendo flexível a mudanças, adequando cada fase e atividade, da melhor forma possível, à realidade de seu negócio.

Uma discussão mais detalhada das sugestões, recomendações e destaques aqui feitos é apresentada ao longo da extensão da dissertação intitulada "Produção de *Software Add-On* – Sugestões de Procedimentos para o Ciclo de Vida do Produto", a seguir.

CAPÍTULO

1 - INTRODUÇÃO

Em vista de um mercado altamente especializado, verifica-se crescente a demanda por novas soluções cada vez mais sofisticadas.

Contudo, a indústria de *software* requer um alto volume de investimentos durante todo o processo de produção do *software*, dificultando a disponibilização final do produto ao mercado.

Desta forma, vários produtos são desenvolvidos e postos no mercado com o objetivo de suprir carências de outros produtos ou propor novas funcionalidades não implementadas pelos fabricantes do *software* original, ou ainda sugerir novos paradigmas para outros produtos de *software*. A produção deste tipo de *software* torna-se mais viável por exigir menores investimentos em todas as etapas de seu processo. Tais produtos são também referenciados, pela indústria, como *software add-on*.

Assim, quem disponibilizar um *add-on* a um *software* comercial já estabelecido, poderá ter excelente diferencial para o acirrado mercado mundial. Além disso, simplifica-se a implantação e manutenção de canais próprios para

marketing, vendas e suporte, pois pode-se aproveitar os canais originais de fornecedores interessados em comercializar o *add-on*.

Como consequência, o *add-on* pode ter um processo de produção rápido e sustentável, o que torna o produtor do *add-on* capacitado em acompanhar a dinâmica do mercado.

A dificuldade, no entanto, reside em se identificar qual *add-on* desenvolver e para qual nicho de mercado. Alia-se, a isto, a ausência de metodologias específicas que englobem todo o processo de produção de *software add-on*.

Portanto, não havendo orientação para acesso ao mercado secundário de *add-on*, é que aqui se propõe um certo grau de formalismo que viabilize, à engenharia de *software*, conceber, desenvolver, preparar e disponibilizar *software add-on*. Para tanto, este trabalho seguirá o molde proposto por [MART93] e expandido por [MOUR94], os quais serão definidos no decorrer do mesmo. Esta tarefa terá como sustentação a análise do processo de produção dos produtos *add-on*, DocM[1] e DocFind[1] , desenvolvidos para serem agregados aos processadores de textos Word[2] e Fácil[3], respectivamente.

1.1 - A Prática da Produção de *Software*

A produção de *software* abrange desde atividades técnicas até gerenciais. Portanto, não se pode descrever a sua prática sem antes concluir que haverá um processo para tal execução.

O processo de produção de *software* parte desta estrutura técnico-gerencial para aplicar o conjunto de métodos, ferramentas e pessoas para a produção de *software*. Mais especificamente, processo de produção de *software*

[1] *Software add-on* DocM e DocFind são de propriedade da Green Software Ltda.

[2] Word é marca registrada de Microsoft Corporation.

[3] Fácil é marca registrada de Fácil Informática Ltda.

é a sequência de passos necessários para se desenvolver e manter um *software* [HUMP95].

Associado a este conceito está a definição de modelo de processo. Este descreve o processo a partir da identificação de regras e funções e através da especificação de tarefas, ou seja, busca formalizar a representação (abstração) dos objetos e atividades envolvidas no processo.

No entanto, vários dos modelos existentes se propõem a formalizar genericamente a produção de *software*, não considerando os diversos segmentos de mercado nem os diferentes níveis de aperfeiçoamento dos processos utilizados pelos produtores de *software*. Como consequência, tem-se que as corporações não se adequam a modelos específicos e estáticos e criam seus próprios métodos para produzir *software*.

Com este desnivelamento, [HUMP88] propôs o modelo CMM (*Capability Maturity Model*), que busca situar o produtor de *software* conforme o nível de formalismo empregado ao processo de produção utilizado. Este modelo posiciona o processo em cinco níveis crescentes de maturidade, partindo do nível inicial, onde preocupa-se apenas com a gerência básica, até o nível mais alto de otimização do processo, no qual o processo está controlado e é continuamente aperfeiçoado.

No entanto, pesquisas realizadas por [BANN91], em 296 grandes projetos que eram desenvolvidos nos Estados Unidos em 1991 mostraram que apenas 2% poderiam estar situados no nível cinco de maturidade, enquanto 88% tinham seus processos pouco formalizados, localizando-se no primeiro nível.

Já no Brasil, a situação se agrava, pois de novembro a dezembro de 1995, pesquisa relatada em [WEBE97] envolvendo 445 empresas do setor de *software* constatou, quanto ao conhecimento e aplicabilidade do modelo CMM, a seguinte situação:

Conhecimento do Modelo Capability Maturity Model (CMM) para Melhoria dos Processos de *Software*		
Categorias	**Nº de empresas**	**%**
Conhece e adota	1	0,2
Conhece e começa a usar	12	2,7
Conhece, mas não usa	49	11,2
Não conhece	377	85,9

Nota: seis empresas não responderam à pergunta.

Tabela 1.1 - Conhecimento do Modelo CMM para Melhoria dos Processos de Produção de *Software* [WEBE97].

Partindo-se desta conjuntura, verifica-se pouco conhecimento do modelo CMM, sugerindo a adoção de modelos mais flexíveis, de forma que os diversos segmentos de mercado possam adequá-los aos seus ambientes.

1.1.1 - Um Molde Realista para o Processo de Produção, Disponibilização e Evolução de *Software*

Analisando os modelos existentes, também identificou-se a atenção nos aspectos técnicos de desenvolvimento de *software*, havendo pouca concentração em questões relevantes para o sucesso do projeto como um todo e que influenciam diretamente nos custos do mesmo.

O processo PDE redefine-se por incluir, além da produção propriamente dita, a disponibilização e a evolução do *software*. Assim, [MART95] e [MOUR97] identificam a produção de *software* comercial composta basicamente pelas atividades de administração, pesquisa e desenvolvimento,

suporte técnico, documentação, manutenção, marketing, vendas e controle de qualidade.

Assim, [MART93] propôs, conforme a Figura 1.1, um "molde representando genericamente a produção de *software*, a partir do qual modelos específicos possam ser construídos e calibrados para casos particulares e realistas do mercado". Posteriormente o trabalho foi extendido, tornando-se um projeto do PROTEM, que passou a referenciá-lo como Molde *R-Cycle*[4] de Produção, Disponibilização e Evolução de *Software* [MOUR94].

O molde está composto por quatro fases, sendo: Concepção, Desenvolvimento, Preparação e Disponibilização, as quais abrangem atividades monofásicas e polifásicas. Estas últimas acompanham o processo durante todas as fases, porém com diferente intensidade.

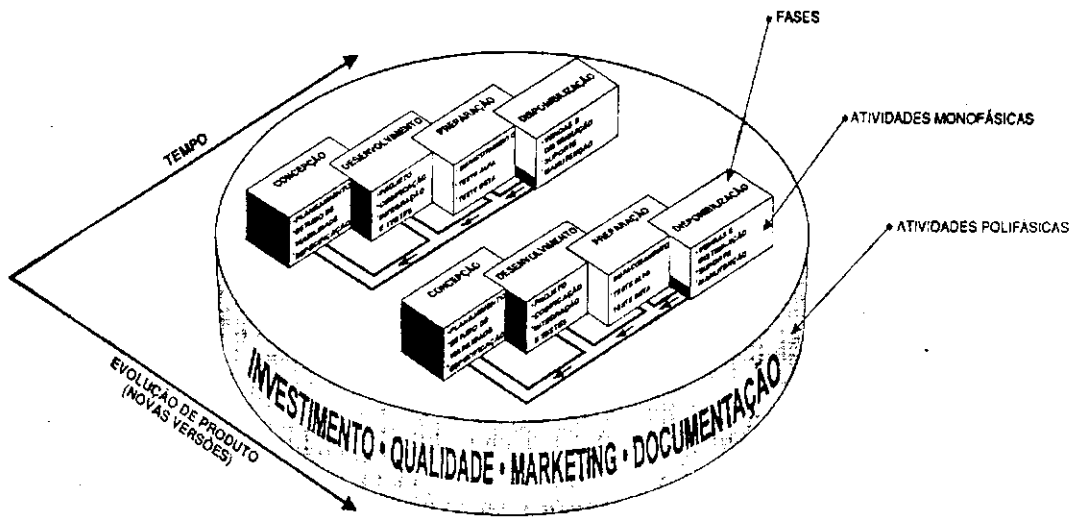


Figura 1.1 - Um Molde Realista para Modelos de Produção de *Software*

Diferentemente dos modelos existentes, este molde busca dar uma conotação mais flexível ao processo de produção de *software*, exigindo que se

[4] Pesquisa financiada com recursos do CNPq/PNUD.

instancie o mesmo para ser aplicado em função das dependências particulares de cada segmento de mercado. Como resultado destas instanciações, tem-se modelos mais específicos, constando de manuais, ou guias práticos, que auxiliam de maneira mais precisa no processo de produção para cada segmento da indústria de *software*.

1.2 - Classificando *Software*

No decorrer da existência dos produtos de *software*, academia e indústria já se empenharam em situar o produto de *software* em categorias específicas. [JONE94], por exemplo, classifica projetos de *software* em seis categorias, indo dos sistemas de gerenciamento de informações e projetos militares, até projetos para técnicos programadores.

No entanto, a dinâmica do mercado não permite a perpetuação de uma classificação em muitos níveis. Sendo assim, realiza-se nesta seção uma tentativa de situar o produto de *software* de acordo com a atual realidade do mercado e, principalmente, com o escopo deste trabalho.

Seguindo a Figura 1.2, permanece em primeiro plano, considerando as duas categorias mais abrangentes, os sistemas de *software* e as aplicações de *software*, as quais acreditamos por bem definí-las.

Na primeira categoria estão inseridos os produtos destinados ao controle de programas, incluindo os sistemas operacionais, *software* de comunicação e gerenciadores de bancos de dados. Já as aplicações de *software* se caracterizam por ser qualquer conjunto de programas que processam dados para o usuário, indo de um *software* de controle empresarial até processadores de textos.

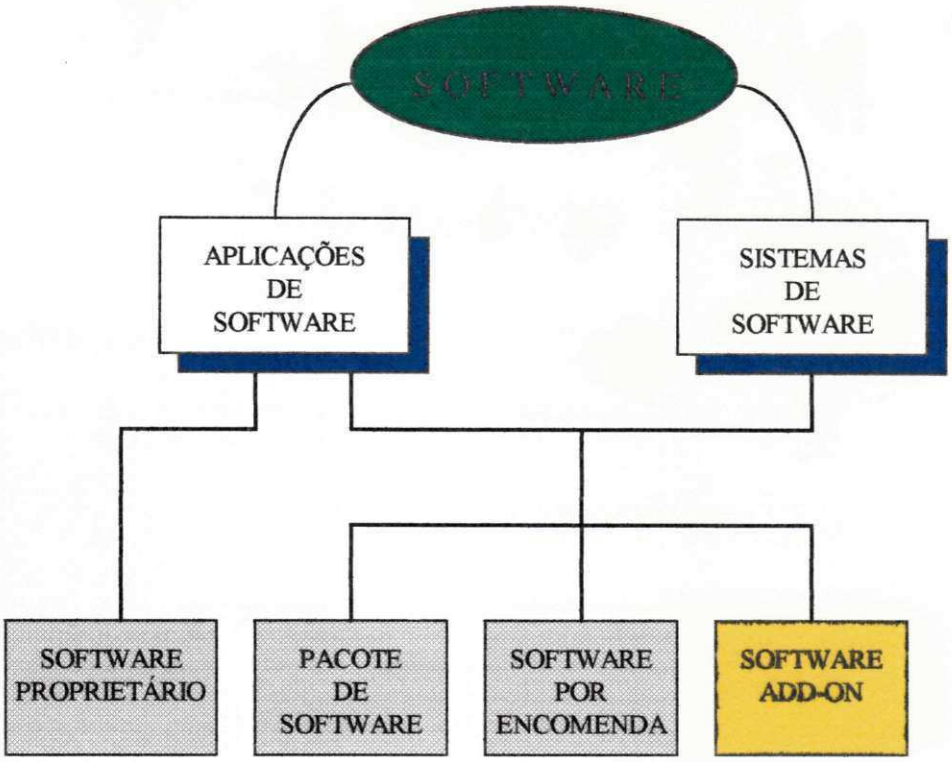


Figura 1.2 - Classificação de *Software* em Categorias.

A partir da Figura 1.2, seguem algumas considerações para os quatro últimos sub-grupos identificados no nível mais inferior:

•*Software* proprietário é aquele desenvolvido para um único cliente. Porém, a diferença está no fato de que ambos, produtor e cliente, estão inseridos na mesma empresa, onde o produtor geralmente é um departamento de desenvolvimento de aplicações e de processamento de dados.

• Como pacote de *software* pode-se definir por um conjunto de programas combinados e que formam uma única aplicação. Podem ainda combinar várias aplicações em um único pacote, podendo ser um gerenciador de banco de dados, um processador de textos com planilhas e gráficos, todos podendo se comunicar entre si. São desenvolvidos para o mercado horizontal de *software*. A indústria referencia o mercado horizontal para os produtos de *software* voltados ao público em geral, independentemente de algum nicho específico.

• Já o *software* por encomenda se caracteriza normalmente por uma relação um-para-um entre produtor do *software* e cliente. Pode, contudo, haver outros clientes, geralmente em quantidade bastante restrita. Verifica-se assim um produto verticalizado, ou seja, voltado a um nicho específico de mercado.

A partir daqui, enfatiza-se a categoria de *software add-on* e, por isso, sua representação gráfica está diferenciada das demais na Figura 1.2.

1.2.1 - *Software Add-on*

No contexto de produto de *software*, tem-se como *software add-on* aquele que, por motivos mercadológicos, é agregado a um *hardware* ou a um outro *software* para ser comercializado, em conjunto, como um único produto [SELE96].

No entanto, isto é uma referência da indústria de *software* e que, devido às diferentes maneiras de se produzir e disponibilizar um *software add-on,* não se encontra formalização terminológica, podendo o mesmo ser denominado plug-in, bundle, add-in, etc.

O termo *add-on* foi usado inicialmente para designar módulos de *hardware*, como placas de circuito impresso, que eram projetados para serem conectados a um *socket* dentro do computador. De forma análoga, tem-se a indústria de *software* utilizando tal conceito para referenciar produtos que tenham como meta agregar/adicionar valor funcional a outros produtos maiores, tornando-os, muitas vezes, mais fáceis de serem usados. A Figura 1.3 simboliza as possibilidades de agregação do *software add-on* em relação ao *software* hospedeiro,

Acredita-se que a terminologia mais expressiva para o conceito de agregação de valor funcional a um determinado *software*, independentemente dos métodos usados para comercialização, seja *add-on*. Além disso, um *software* também se caracteriza como *add-on* quando sugere um novo paradigma ou forma de execução para uma funcionalidade existente em outros

produtos de *software*, como por exemplo uma interface diferenciada ou até mesmo uma maneira melhor ou mais fácil de montar e emitir um relatório.

Identifica-se assim o processo de produção de *software* a ser focado nesta dissertação, sendo este o termo utilizado no decorrer da mesma. Vale salientar também que, para esta categoria de *software*, podem ser consideradas mais duas subdivisões, conforme sua disponibilização: *software add-on* horizontal e vertical.

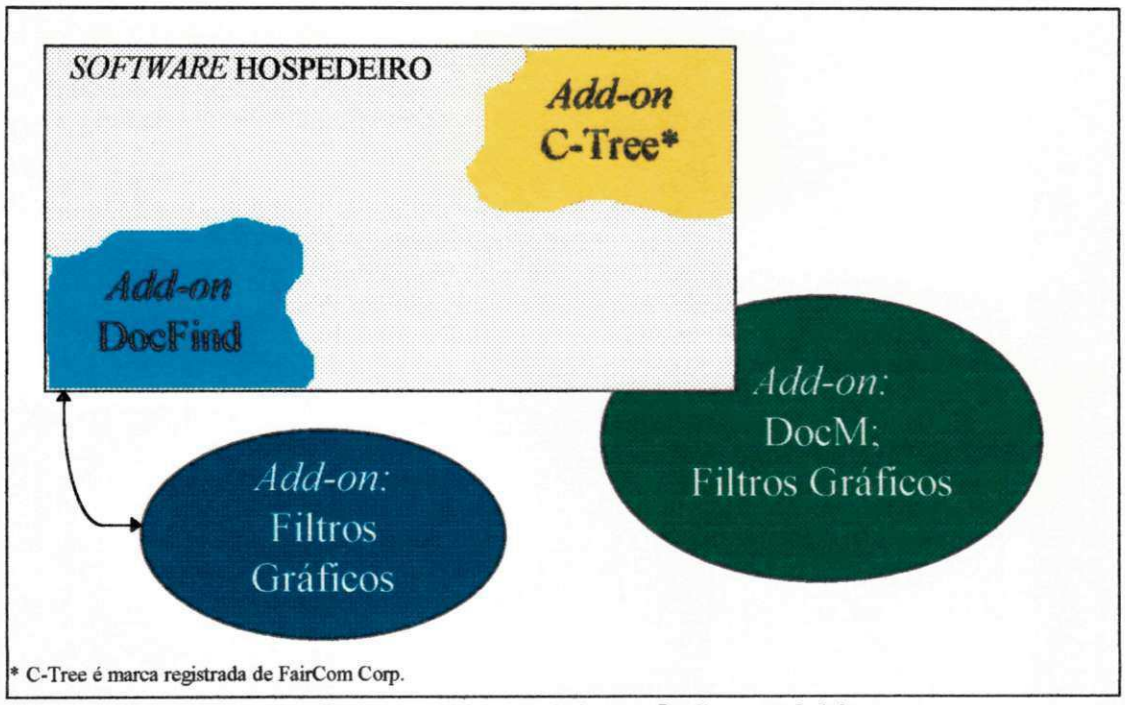


Figura 1.3 - Relação *Software* Hospedeiro e *Software Add-on*

A Figura 1.3 situa o *software add-on* em relação ao *software* hospedeiro e conforme sua agregação a este, tanto no que se refere aos aspectos técnicos quanto comerciais.

Portanto, o *software add-on* pode estar diretamente inserido no hospedeiro, sendo disponibilizado ao mercado de forma conjunta (*add-on* DocFind e C-Tree). O *add-on* pode ainda ser comercializado de forma independente, geralmente no caso de não precisar ser ligado (linkeditado) ao código-fonte do hospedeiro (DocM e Filtros Gráficos).

1.3 - Enfoque da Dissertação

Considera-se, a princípio, a Figura 1.1, que expressa a estrutura do molde R-Cycle e do processo PDE de *Software*. Sabendo-se ser um molde que permite sua extensão a diversos segmentos do mercado, verifica-se que a instanciação tem como função calibrar as fases que compõem o molde para cada caso particular da indústria de *software*.

Assim sendo, o enfoque desta dissertação concentra-se em analisar e organizar, sob o ponto de vista do molde, as principais atividades a serem desempenhadas para a produção e disponibilização de *software add-on*.

Partindo-se da classificação de *software* exposta na seção 1.2, realiza-se a segmentação do mercado-alvo, procurando identificar oportunidades para a produção de um *software add-on* que sirva de auxílio, como estudo-de-caso, para este trabalho.

Desta forma, a dissertação está embasada na análise do processo de produção de dois produtos de *software add-on*, ambos da empresa Green *Software* Ltda., sediada nesta cidade. O primeiro trata-se do *add-on* DocM, projetado, e inicialmente implementado, para ser agregado ao processador de textos Word, da Microsoft. Já o segundo, denominado de DocFind, foi preparado para se unir ao processador de textos Fácil, da empresa de mesmo nome e com sede em Blumenau/SC.

Sendo assim, a meta do trabalho se concentra em analisar, identificar e relatar com que intensidade deve ser empreendida, para *software* do tipo *add-on*, cada uma das quatro principais fases do processo PDE, além das atividades de acompanhamento (atividades polifásicas). Busca-se assim descrever o ciclo de vida para *software add-on*, parametrizando a sequência de passos para que se produza este tipo de *software* de maneira mais eficiente e com menores riscos.

1.4 - Contribuições da Dissertação

A dissertação formará uma documentação sobre as alternativas e parâmetros que devem receber maior atenção quando da produção de um *software add-on*, identificando as dificuldades existentes ao longo de todo o processo, desde a concepção do produto até a sua oferta ao mercado.

Esta documentação também tem um cunho prático, quando se considera o resumo executivo. Este contribui, à academia e à indústria, como um guia prático e flexível, sugerindo diretrizes a procedimentos ideais, a serem seguidos quando da intenção de se analisar o mercado-alvo e disponibilizar a este um *software add-on*.

Segundo [MELO97], para o programa SoftEX-2000, os fatores para aumento da competitividade da indústria brasileira devem estar voltados às empresas de pequeno porte. Para estas, a sobrevivência no mercado se sustenta a partir de dois tipos de estratégias: o primeiro consiste na "estratégia de nicho" e o segundo pode ser denominado "estratégia de interstício". Nesta última, por sua vez, o caráter multidimensional dos produtos de *software* é aproveitado para a implementação de uma diferenciação de produto voltada para a ocupação de pequenos espaços, mas que representam um mercado de grandes proporções, deixado pelas empresas líderes, cujas linhas de produtos jamais podem ser amplas o suficiente para oferecer todas as variedades possíveis.

O *software add-on* situa-se na segunda estratégia, estando em plena sintonia com as perspectivas do programa SoftEX-2000, contribuindo para o mesmo e para o aumento da competitividade da indústria brasileira.

Além disso, vislumbra-se fornecer informações preciosas também ao projeto Genesis, o qual busca apoiar a formação de novos empreendedores no setor de informática [GENE97]. Tais empreendimentos se caracterizam inicialmente como pequenos negócios, estando assim em sintonia com SoftEX-2000 e com projetos de *software add-on*.

Assim sendo, o presente trabalho trará à academia, e ao mercado produtor de *software*, uma parametrização das atividades inerentes ao processo de produção, evolução e disponibilização de *software* do tipo *add-on,* de maneira que se consiga iniciar tal processo com o mínimo de controle gerencial para se chegar a um software add-on comercializável.

Espera-se, por fim, que esta documentação, ao colaborar para o processo de inovação tecnológica, apropriando conhecimentos para transferir aos empreendedores, seja de importância para o segmento da indústria que almeja abarcar este mercado alternativo de *software*.

1.5 - Organização da Dissertação

A estrutura do presente trabalho está baseada na abordagem do modelo do processo de produção proposto por [MART93]. Não requerendo para isto muita complexidade na organização geral, expõe-se a seguir sua estrutura, passando em seguida a uma breve descrição dos pontos a serem detalhados em cada capítulo.

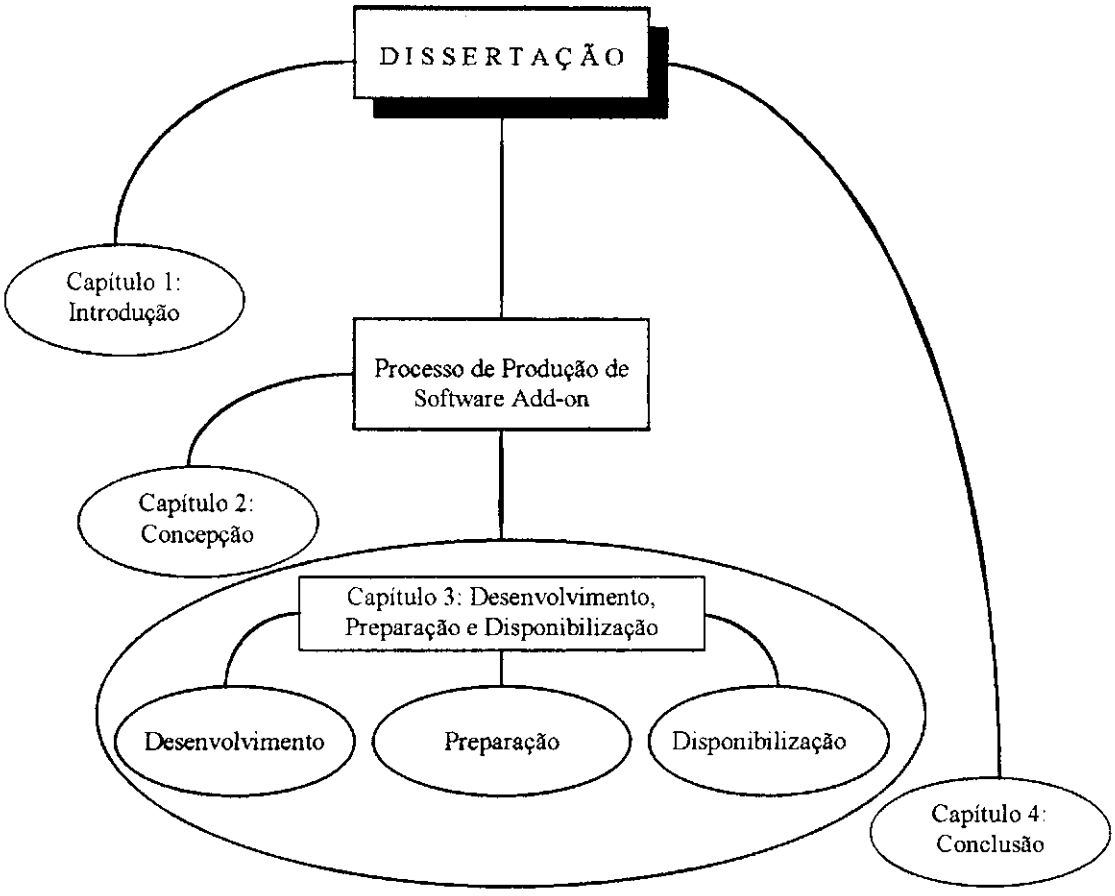


Figura 1.4 - Estrutura de Organização da Dissertação

O Capítulo 2 se caracteriza por identificar a primeira etapa do processo de produção de *software add-on*: a fase de concepção. São abordados aspectos de viabilidade técnica e comercial, com levantamento dos segmentos de mercado. A partir daí são identificadas as funcionalidades a serem acrescentadas a um *software* hospedeiro, bem como os métodos de interface a serem adotados para com este.

A estrutura da dissertação proporciona um capítulo exclusivo à fase de concepção e, as três fases restantes do processo de produção (desenvolvimento, preparação e disponibilização) são concatenadas no Capítulo 3. Isto se justifica pela importância da fase de concepção na estratégia para produção de *software add-on*. É o ponto chave para a decisão de continuar ou não com a produção do mesmo.

O início do Capítulo 3 compreende a parametrização da fase de desenvolvimento (seção 3.1), onde são analisadas as atividades de implementação das funcionalidades especificadas na fase de concepção. Avaliam-se também, aspectos técnicos de integração e testes dos produtos em questão. Em seguida, descreve-se a fase de preparação do produto (seção 3.2), que envolve as atividades de teste e aceitação. No caso do *add-on* são redefinidas, ou confirmadas, as estratégias de comercialização e empacotamento, pois na etapa de concepção algumas medidas a este respeito já devem ter sido tomadas.

Para encerrar o processo de produção chega-se à etapa de disponibilização do produto. Deve ser posta em prática a etapa final para comercialização do *add-on,* confirmado-se os acordos para distribuição, suporte e manutenção dos produtos. Esta etapa é discutida na última seção do Capítulo 3 (seção 3.3).

Salienta-se que em todas as fases poderão constar parâmetros relativos a investimentos, controle de qualidade, marketing e documentação, além de aspectos de interação com parceiros comerciais. Estes parâmetros formam as atividades polifásicas definidas no molde R-Cycle de produção, disponibilização e evolução de *software.*

Para efeito de ilustração e exemplificação, serão constantemente referenciados, no trabalho, dados relacionados aos dois estudos-de-casos considerados. Acreditou-se por bem não criar um capítulo à parte para os estudos-de-casos para que, através de dados reais, fosse instanciado o molde, tornando mais límpido e prático o processo de produção do *add-on.*

Para concluir, no Capítulo 4 são feitas as considerações finais a respeito do modelo proposto. São resumidamente descritas as fases do modelo e as atividades envolvidas em cada fase. Por fim, são levantadas as possíveis expansões ao modelo em decorrência da dinâmica tecnológica e de mercado.

PROCESSO DE PRODUÇÃO DE *SOFTWARE ADD-ON*

CAPÍTULO

2 - CONCEPÇÃO

Esta fase inicial do processo marca o ponto onde é idealizado o *software* a ser produzido e onde são estabelecidas as estratégias para a execução do projeto.

A fase de concepção pode ser disparada pela descoberta de oportunidades de mercado, por solicitações de clientes, por contratos com parceiros ou pela evolução desejável de um *software* já disponível [MART93].

Apesar de vários tipos de *software* poderem ser influenciados por estes motivos, verifica-se que o *add-on*, em especial, está diretamente dependente do último, pois necessita de outro(s) *software*(s) para existir e prosperar no mercado.

Sendo assim, a instanciação desta fase, para o *software add-on*, requer necessariamente como atividade inicial que sejam analisados os segmentos de mercado, procurando o nicho de mercado a ser almejado, identificando riscos do projeto e levantando-se os aspectos de viabilidade do mesmo.

Para acompanhar as atividades envolvidas nas fases do processo, as figuras correspondentes a cada fase conterão, em cada atividade, apenas as particularidades para *software* do tipo *add-on*.

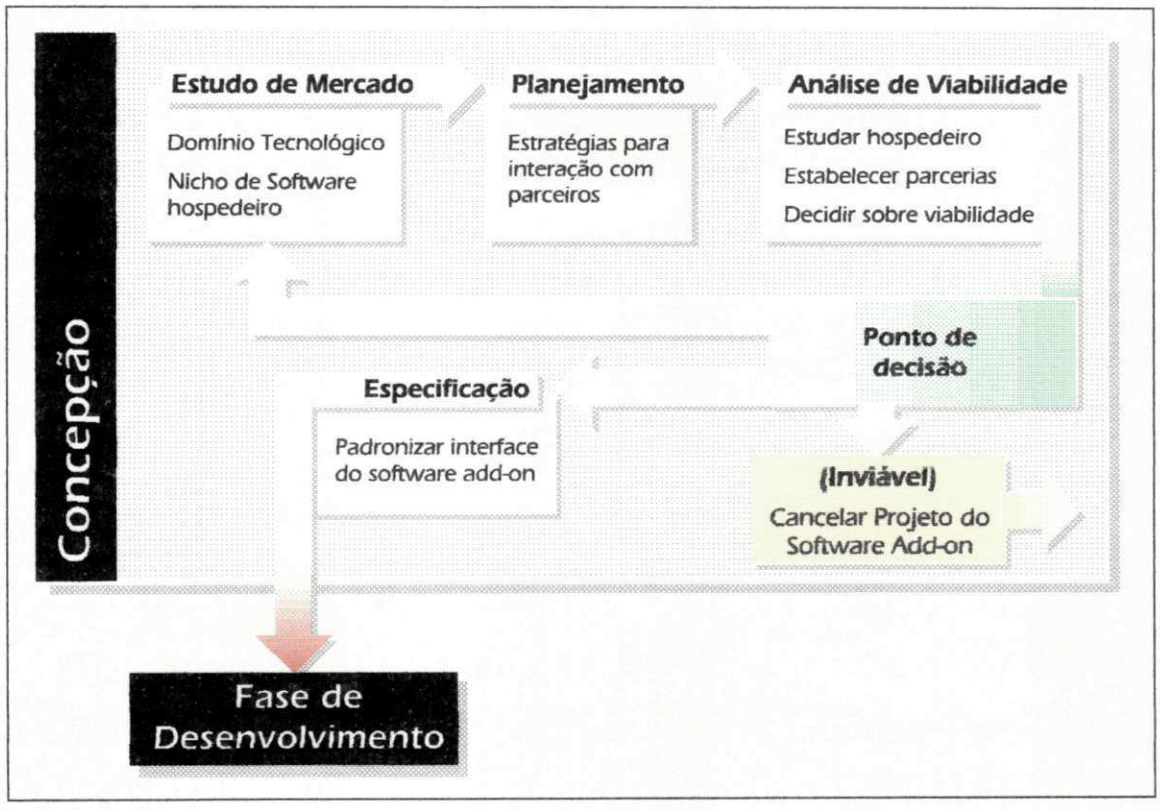


Figura 2.1 - Síntese das Atividades para a Fase de Concepção

2.1 - Levantamento dos Segmentos de Mercado

Pode-se considerar este passo como fundamental para o posicionamento estratégico frente à decisão de se iniciar ou não a produção de um *software*. Primeiramente, no caso do *add-on*, por ser necessário a existência de um outro *software* - geralmente um pacote - e, em seguida, porque a atividade de identificar este *software* hospedeiro não se faz tão trivial.

Dentro deste contexto, alguns fatores são proeminentes para a definição do nicho de mercado a atuar. Realizando-se uma auto-análise, deve-se partir da dimensão do empreendimento do produtor do *software add-on* e

identificar qual a sua tecnologia dominante, e, esta é que indicará o caminho para o levantamento de mercado.

Em ambos estudos-de-casos, realizados a partir de produtos da empresa Green Software[3], utiliza-se da tecnologia aplicada em outro produto da mesma, denominado Light Text, e que se caracteriza por ser uma biblioteca orientada a objetos de apoio à recuperação textual [BUSS95]. Neste ponto, deve-se estabelecer o tipo de aplicação que se pretende produzir.

No caso da empresa Green, o objetivo foi disponibilizar uma ferramenta que auxiliasse a localização dos arquivos, criados pelo usuário em processadores de textos, e que estivessem armazenados no disco rígido do computador, usando para isto a tecnologia de recuperação textual.

A partir de tal objetivo, os processadores de textos tornaram-se alvos de novas funcionalidades (a serem oferecidas a partir da tecnologia de recuperação textual da empresa Green).

Tendo sido segmentado este mercado, é traçada a busca por oportunidades funcionais e são identificadas a demanda e a satisfação dos usuários em relação a este mercado. Devem ser identificados também o volume de vendas e a base instalada de cada produto, sendo de extrema importância à análise de viabilidade comercial. Posteriormente se pode constituir por definitivo qual o nicho de *software* hospedeiro a ser atacado.

2.1.1 - Identificando Oportunidades

Considera-se, inicialmente, a análise do conjunto dos produtos que estão inseridos no mercado segmentado (processadores de textos) e que estão melhor cotados no mesmo.

[3] Green Software Ltda: empresa voltada aos segmentos de recuperação textual e multimídia, estabelecida no Parque Tecnológico de Campina Grande / PB.

Já tendo sido caracterizado o tipo de ferramenta a desenvolver, poder-se-á então realizar uma análise minuciosa, mas específica, voltada à descoberta de lacunas nestes produtos hospedeiros e que, eventualmente, possam ser preenchidas por um *software add-on*, avaliando-se a usabilidade destes produtos. Para tanto, deve-se utilizar o método de testar a usabilidade dos produtos hospedeiros, sendo que algumas das características envolvidas neste método citam a busca por problemas de usabilidade e, a partir da análise dos resultados dos testes, pode-se ter um diagnóstico recomendando mudanças e correções para os problemas encontrados. Para uma documentação mais completa sobre testes de usabilidade, ver [AZEV96] e [MOUR97].

Portanto, os testes de usabilidade além de auxiliarem na descoberta de oportunidades, são uma importante ferramenta de marketing desde a fase de concepção, pois permitem reduzir os riscos de insucesso do produto no mercado, ou seja, possibilitam que seja identificada a real necessidade de um *add-on* para o nicho de mercado visado.

Como processadores de textos estabelecidos no mercado, identificou-se o líder como sendo o Word para ambiente Windows. Seguindo-o, porém oriundos de produtores nacionais, estavam o processador Fácil, da empresa de mesmo nome, e o Carta Certa, atualmente pertencente à DTS Informática.

Dentre eles, todos eram passíveis de receber um *add-on* com as características vislumbradas, pois o processo de localização de documentos, implementado por tais processadores, deixava a desejar.

Contudo, isto era apenas um sentimento do produtor de *software*. Para não permanecer no empirismo, deve-se realizar pesquisa de mercado, buscando reduzir riscos. Assim, questionou-se o público usuário quanto às suas expectativas em relação a estes produtos e quanto às funcionalidades de interesse.

2.1.2 - Questionando o Público Alvo

Esta atividade visa obter o *feedback* dos usuários dos produtos de *software* segmentados no item anterior. Os questionamentos devem ser inerentes aos interesses do estudo em questão sem, contudo, explicitá-los ao público a ser entrevistado e, este público, deve ser selecionado dentre diferentes áreas profissionais, buscando-se não tendenciar os resultados da pesquisa.

Assim, demarcou-se o questionário com uma explicação sucinta sobre o objetivo do mesmo e distribuiu-se o conjunto de questões em quatro grupos distintos, porém inter-relacionados, como segue:

- O primeiro grupo tratou de posicionar o usuário quanto ao seu perfil, identificando sua área de atuação e sua amigabilidade com o ambiente computacional utilizado.
- O segundo e o terceiro grupos abordaram os aspectos referentes ao segmento de *software* que está sendo visado. Estes grupos de questões foram definidos como:
  - "Sobre editores de textos";
  - "Sobre editores de textos para Windows".
- Na quarta parte do questionário afunilaram-se as questões para características referentes às funcionalidades que pretende-se adicionar através do novo *software add-on*. Assim sendo, foram levantadas questões que permeavam a satisfação dos usuários quando da necessidade de organizar e localizar seus documentos. A este conjunto de questões denominou-se "Sobre organização e localização de arquivos".

Para um detalhamento do questionário, ver Apêndice A.1.

Após o recolhimento dos questionários, chega-se à etapa de tabulação dos dados adquiridos, buscando gerar informações relevantes à

tomada de decisão sobre a produção do *add-on*. Isto é possível através da análise dos dados pertencentes, principalmente, ao quarto grupo de questões, que fornece subsídios referentes à necessidade de tais funcionalidades serem implementadas.

A amostra considerada para estudo reflete mais especificamente o ambiente universitário, englobando tanto servidores como estudantes. No entanto, buscou-se questionar usuários sem envolvimento com profissões técnicas de informática, objetivando não influenciar os resultados. Assim, a amostra compreendeu 113 usuários questionados. Apesar de pouco expressivo em relação ao volume de usuários do *software* hospedeiro a ser focado, esta pesquisa conseguiu o *feedback* dos mesmos, identificado suas realidades e perspectivas para com o ambiente computacional utilizado.

Da amostra considerada, verificou-se que 103 entrevistados utilizam o ambiente operacional Windows e, destes, 101 utilizam algum processador de textos.

Buscando razões para empreender o projeto do software add-on, identificou-se ainda que, destes 101 usuários, 81 não preenchem os resumos informativos de seus documentos, 16 acham difícil a localização dos mesmos e 35 prefeririam desconhecer a estrutura de arquivos e diretórios para que pudessem localizar seus documentos. O Gráfico 2.1 identifica também o desejo dos usuários em localizar com maior facilidade os seus documentos de forma a torná-los mais produtivos.

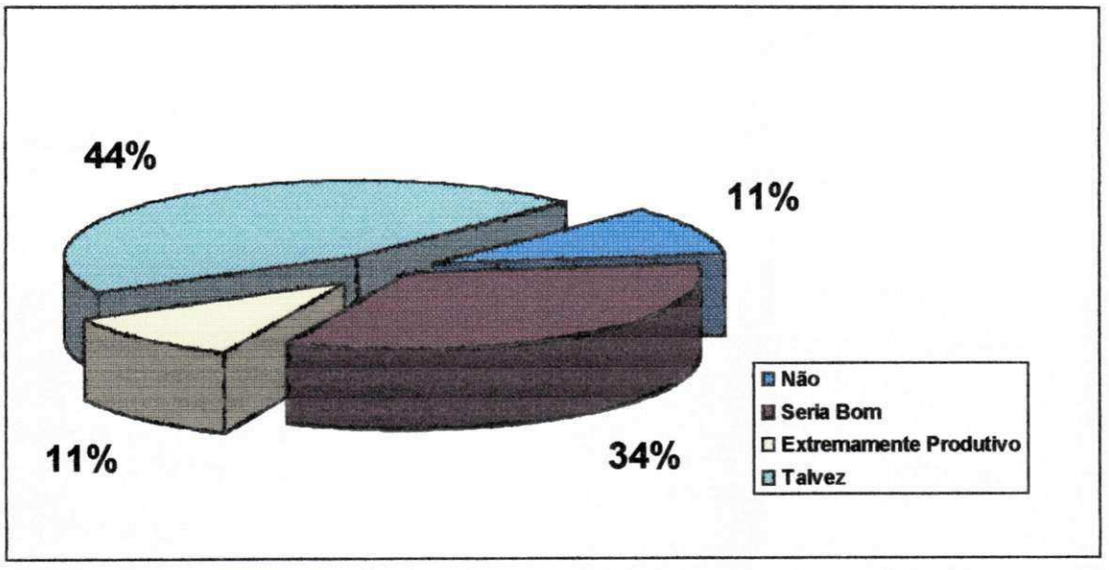


Gráfico 2.1 - Resultado de Pesquisa para Usuários com Intenção em Recuperar Documentos pelo Conteúdo.

2.1.3 - Estabelecimento do Nicho de *Software* Hospedeiro

A decisão de se estabelecer qual o nicho/mercado-alvo de *software* a ser atacado, é auxiliada pela verificação dos dados oriundos das questões dos grupos dois e três do questionário, mais especificamente no que tange ao *software* utilizado pelos usuários.

Desta forma, o Gráfico 2.2 considera somente os usuários do ambiente Windows e que usam algum processador de textos. Confirma-se, assim, o sentimento inicial de enfocar, como *software* hospedeiro, o processador de textos Word for Windows.

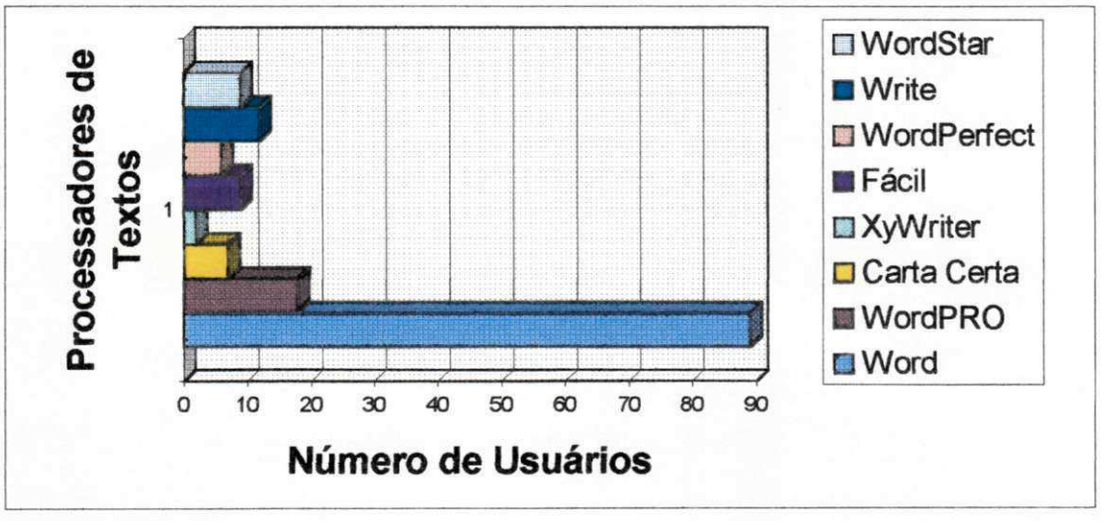


Gráfico 2.2 - Resultado de Pesquisa para Número de Usuários por Processador de Textos

No entanto, poder-se-ía definir um alvo mais abrangente, onde constasse mais do que apenas um *software* hospedeiro. Assim sendo, esta seção, juntamente com a anterior, também auxiliam no caso de um possível retorno para redefinição do foco de atuação.

2.1.4 - Análise de Requisitos

Tendo-se obtido o *feedback* do mercado, buscar-se-á com a análise de requisitos identificar os problemas do mercado e dos clientes que se quer solucionar. Assim, o documento deve abranger os pontos críticos que o *software add-on* terá como objetivo resolver.

"O estudo dos requisitos pode ser executado pelo cliente, desenvolvedor, pela equipe de marketing, ou qualquer combinação destes três" [GHEZ91].

Os requisitos devem abordar os problemas do mercado e não as soluções para estes. Portanto, deve haver informação suficiente nos requisitos do produto para gerar posteriormente o documento de objetivos do mesmo - no projeto arquitetural - a partir apenas desta informação [MOUR97].

Por fim, no caso de software do tipo add-on, o documento gerado deverá servir como ferramenta à equipe responsável em argumentar, junto aos eventuais parceiros, a importância em solucionar os problemas identificados.

2.2 - Planejamento

Possivelmente a atividade de planejar o projeto pode ser requerida já como o primeiro passo da fase de concepção de um *software*. No entanto, por motivos metodológicos, acredita-se ser necessário antes definir o nicho de *software* a se investir, bem como saber que requisitos o *software add-on* deverá atender. Somente assim poder-se-á iniciar o planejamento do projeto, buscando parametrizar a operacionalização do mesmo.

Esta atividade de planejamento, dentro do enfoque de produção de *software add-on* busca, além de estimar prazos e custos, organizar a estrutura que será adotada para a execução do projeto.

Assim sendo, posiciona-se o presente trabalho de acordo com [HUMP95], o qual descreve que, além de estimativa de custos, cronogramas e captação de recursos, é necessário também organizar a estrutura operacional e, principalmente, registrar o que foi inicialmente submetido e o que realmente se executou.

Adequando-se estes requisitos do planejamento de um projeto, à realidade de um *software add-on*, verifica-se como princípio desta tarefa traçar a operacionalização das demais atividades intrínsecas à fase de concepção. Isto ocorre porque, diferentemente de pacotes de *software*, o processo de produção de *add-on* requer o controle de riscos, no estudo sobre demanda e viabilidade, e maiores investimentos nesta fase de concepção e no desenvolvimento.

Portanto, organizar como se darão os estudos de viabilidade, e definir como será a interação entre os parceiros, torna-se o principal objetivo do planejamento para *software add-on*, além é claro das estimativas de custos e prazos, inerentes a qualquer plano de execução de projetos.

Tendo-se disponível o documento que descreve os requisitos para o *software add-on*, o primeiro passo é definir as equipes responsáveis pela análise de viabilidade técnica e comercial. Determinam-se também, para atuar em paralelo, as pessoas que irão tratar de definir as nuances relativas à interação entre produtos e empresas.

A partir daí, os próximos aspectos do planejamento se concentram em prazos e custos. Estima-se o processo de especificação, bem como as atividades existentes nas demais fases do processo de produção, delegando pessoas às equipes e definindo metas para as mesmas.

2.3 - Análise de Viabilidade

Produtos de *software* do tipo *add-on* se caracterizam por serem altamente dependentes de fatores externos. Haja visto a necessidade da existência de outro *software* para se agregar e, consequentemente, de outro foco de mercado para se estabelecer uma parceria, a análise de viabilidade torna-se vital para a produção do *add-on*. Assim sendo, aqui é o momento de se decidir a respeito da continuidade ou não do projeto para o *software add-on*.

Sob esta esfera, torna-se mais enfático dividir esta atividade de acordo com os aspectos técnicos e comerciais, buscando-se alternativas para integração, as quais servirão de suporte para se estabelecer a interação com o parceiro, conforme descrito na seção 2.4.

2.3.1 - Viabilidade Técnica de Integração de *Software*

O procedimento inicial desta atividade se concentra em identificar aspectos favoráveis à integração técnica entre os dois produtos de *software*. Para isto, torna-se necessário um estudo minucioso do *software* hospedeiro no que tange às suas características de "sistema aberto", ou seja, de que maneira tal *software* permite que outros se agreguem ou interajam com ele.

É aconselhável que, além de identificar tais alternativas, seja também descrita a dependência para com o produtor do *software* hospedeiro. Sugere-

se, para tanto, que sejam expostos os prós e contras para cada solução vislumbrada.

Como vantagens pode-se considerar um *software* que disponibilize um conjunto de funções, formando uma API (*Application Program Interface*), ou o recurso de macros, como foi o caso encontrado no processador de textos Word.

Já os aspectos inviabilizadores podem consistir dos resultados dos testes de usabilidade, que podem condenar mudanças no hospedeiro, ou de não se ter o formato de alguns arquivos manipulados pelo *software* hospedeiro ou ainda, o que é mais crítico, verificar a necessidade de ter que interagir com o código-fonte do mesmo. Em ambos os casos será imprescindível o contato com o produtor do *software*, visando obter esclarecimentos e/ou soluções.

Desta forma, o documento gerado auxiliará no processo de interação com a equipe técnica, responsável pela integração dos produtos e, desta, para com os desenvolvedores do *software* hospedeiro.

Vale salientar que nos estudos-de-casos realizados, não houveram aspectos que inviabilizassem tecnicamente o projeto.

2.3.2 - Viabilidade de Parceria Comercial

Por se tratar da produção de *software add-on*, as estratégias de comercialização de produtos prontos (*software* de prateleira) não estão em foco nesta seção, mesmo que estes se constituam em *software* de agregação de valor funcional. Todavia, não está descartada a possibilidade do *software add-on* ser comercializado isoladamente.

Partindo-se do estudo de viabilidade técnica, identificam-se aqui as possíveis alternativas para se estabelecer uma parceria comercial de forma a disponibilizar, de forma integrada, ambos os produtos. Os aspectos destas negociações estão melhor detalhados na seção 2.4.

O estudo de viabilidade também deve se preocupar com os prazos e funcionalidades das *releases* a serem disponibilizadas pelo produtor do *software* hospedeiro, bem como com os produtos concorrentes. Evita-se assim que seja dado início ao processo de produção de um *add-on* ocioso e inviável comercialmente.

No entanto, são os aspectos comerciais que mais provavelmente venham a inviabilizar o projeto do *add-on*. Os motivos podem ser diversos, onde cita-se a falta de interesse mercadológico por parte do produtor do *software* hospedeiro ou a dificuldade de se conseguir estabelecer contato com o mesmo.

Neste ponto tornou-se inviável o *add-on* para o processador Word, sendo descartado o projeto do DocM e sua produção cancelada. O motivo para tal decisão foi a dificuldade existente para uma empresa brasileira de *software* iniciar qualquer negociação junto a empresas de porte como Microsoft, além dos aspectos geo-econômicos.

Apesar disso, continuaram-se com as especificações do produto e, por razões metodológicas, são abordadas em seções adiante pela contribuição técnica prestada ao enfoque do *software add-on*.

Desta forma, deve-se conhecer as expectativas do outro produtor, tendo sua posição quanto ao interesse em receber as funcionalidades propostas. Isto tornou-se mais viável junto a empresas nacionais, tendo a Fácil Informática lançado seu aval positivo em relação ao interesse com o projeto do *add-on* DocFind.

2.4 - Interação com o Parceiro

Chega-se a esta etapa com um conjunto importante de informações a respeito da viabilidade do *add-on*. Estas servirão como guia, ou suporte para esclarecimentos de dúvidas, relativos a este processo de interação com o(s) produtor(es) do *software* hospedeiro.

Divide-se também esta atividade em aspectos técnicos e comerciais. Isto deve-se à especificidade dos documentos gerados anteriormente, e para permitir que se formem equipes distintas para interagirem com os respectivos grupos da outra empresa.

2.4.1 - Aspectos Técnicos

Tem-se o primeiro contato com o segmento de mercado escolhido já no momento de se identificarem oportunidades, quando da aquisição de uma cópia do *software* hospedeiro. Outro momento é quando da análise de viabilidade comercial, pois interage-se com o produtor buscando conhecer seus interesses pelas funcionalidades ora propostas.

No entanto, as atividades descritas nesta seção aproximam mais os projetistas de ambas empresas. São discutidos assuntos mais específicos de desenvolvimento, cabendo aos técnicos do *software add-on* propor alternativas de integração entre os produtos, expondo suas dúvidas, buscando soluções e melhores estratégias de ligação entre os mesmos.

Possivelmente voltar-se-á a esta atividade durante o decorrer do processo. Contudo, antes de iniciar a especificação funcional propriamente dita, deve-se sair com algumas posições definitivas. Entre elas citam-se:

- As plataformas de desenvolvimento usadas pelos dois produtos;
- O tipo de interface que será estabelecida entre os produtos de *software*;
- Os momentos em que as funcionalidades do *add-on* serão executadas pela aplicação que o recebe;
- O tipo de interface de usuário solicitada;

No caso estudado junto à empresa Fácil, o primeiro contato trouxe o posicionamento favorável às funcionalidades propostas, bem como informações técnicas específicas a respeito do *software* hospedeiro e de uma possível integração.

2.4.2 - Aspectos Mercadológicos e Comerciais

Paralelamente à proposta das funcionalidades do *software add-on*, a equipe de marketing terá o contato inicial, junto à empresa parceira, com relação a aspectos mercadológicos e comerciais. Como colocado na seção 2.3.2, devem ser obtidas informações a respeito das pretensões do parceiro em implementar ou não tais funcionalidades, bem como das suas perspectivas de prazos para a liberação das mesmas.

Considera-se ainda, que a análise de viabilidade tenha certificado como possível a integração, podendo-se disponibilizar ambos os produtos de forma conjunta, como um único pacote de *software*. Mesmo por este ângulo, verificam-se oportunidades diferentes para a comercialização dos produtos como, por exemplo, a incorporação da tecnologia desenvolvida pela empresa criadora do *software add-on* ou o direito à exclusividade do uso do mesmo. Assim, deve-se considerar os aspectos legais inerentes a cada alternativa existente. O produtor do *software* hospedeiro fará sua opção ao ter a real noção do diferencial trazido ao seu *software*.

Desta forma, cabe aqui utilizar das possibilidades emergentes, buscando um acordo razoável que viabilize a parceria comercial e o início da especificação do *software add-on*. Assim, a proposta inicial visa estabelecer uma relação para o uso do *software add-on* agregado ao produto alvo, estando os produtores deste comprometidos com a outra empresa através de *royalties*.

Há, contudo, variações para este acordo, como por exemplo a disponibilização condicional do *add-on*, dependendo da solicitação dos clientes, que podem optar ou não pela presença do *add-on*. Porém, análogo aos aspectos técnicos, a interação comercial se prolongará até o final do ciclo do *add-on*, quando estas variações poderão ser lançadas ou enquanto este *software* solicitar manutenção.

2.5 - Especificação

Na seção 2.1 consta o último item que propõe a análise de requisitos, preliminar à especificação e que, neste momento, servirá como ponto de partida para a definição dos objetivos do produto *add-on* a ser desenvolvido. Forma-se, assim, o contexto para as duas divisões iniciais desta fase, as quais estão em sintonia com [GHEZ91]. Este propõe a divisão desta especificação em duas subfases, sendo: projeto arquitetural, ou de alto nível, e o projeto detalhado. Este último correspondendo ao item relativo à especificação dos módulos e funcionalidades do sistema.

O restante da especificação fica por conta da padronização da interface a ser adotada pelo *software add-on*. Em seguida pode-se estabelecer algumas propriedades fundamentais que deverão constar no *software add-on* e que encerram a especificação funcional.

Busca-se assim formar um documento que permita a entrada à fase de desenvolvimento, a qual inicia com as atividades de especificação mais voltadas à implementação do produto em questão.

2.5.1 - Projeto Arquitetural: Objetivos do Produto

A partir da análise de requisitos, este novo documento se concretizará em descrever os objetivos do projeto, estabelecendo vantagens e dando uma visão geral dos produtos envolvidos. São citados a plataforma operacional e o(s) *software*(s) hospedeiro(s) visados.

Desta forma, pode-se relatar as principais características do *software add-on*. Cita-se a estratégia de integração entre os produtos, definindo os módulos macros que formarão o *software* e como estes irão interagir para alcançar os objetivos propostos. Também deve-se estabelecer o tipo de interface com o usuário a ser adotada, quando houver uma, e quando esta interface será disparada.

Abaixo situam-se os aspectos básicos da especificação para os estudos-de-casos em questão. Este tipo de abordagem se dará, no decorrer da dissertação, quando for necessário para uma melhor compreensão dos produtos envolvidos e para melhor especificar as tarefas a desempenhar.

2.5.1.1 - Caso do *Add-on* para Processador Word (DocM)

1 - Objetivos / Overview:

- Oferecer ao usuário Word for Windows a facilidade de localizar seus arquivos através de consultas sobre o conteúdo dos documentos, nome dos arquivos, datas (criação e atualização) e resumos informativos;
- Disponibilizar uma interface gráfica para pesquisa e recuperação de todos os documentos do usuário.

2 - Características:

- Primeira versão deve disponibilizar funcionalidades básicas e que não atrasem a sua implementação, mas que no entanto não venham a comprometer futuras incrementações ao produto;
- Módulo1 - Indexação: a máquina de indexação, que provê a principal característica do produto - pesquisas - estará sendo executada como uma DLL;
- Módulo 2 - Pesquisa e Recuperação Textual: a "cara" do produto somente aparecerá nos momentos de abertura e pesquisa de arquivos e na seleção de novos arquivos a serem indexados. Para isto a interface deverá seguir as tendências do mercado de processadores de textos (Word) para Windows;
- A invocação do produto estará vinculada às funcionalidades de menu e/ou barra de ferramentas do próprio editor de textos em questão, sendo disparada a partir de macros.

2.5.1.2 - Caso do *Add-on* para Processador Fácil (DocFind)

1 - Objetivos/Overview:

- Disponibilizar aos técnicos (analistas/programadores) da empresa Fácil Informática, uma ferramenta de interface de programação (*API - Application Program Interface*) que lhes possibilite implementar módulos para recuperação textual dentro do principal produto desta empresa: o processador de textos Fácil.
- A interface de programação busca oferecer um conjunto de funções, formando assim uma biblioteca para programação. Desta forma, não se tem como objetivo a implementação da interface para o usuário do processador Fácil. Nota-se também que a API será implementada em linguagem de orientação a objetos, porém isto ficará transparente para o usuário (técnico), pois as funções da API estarão sendo disponibilizadas de maneira procedural, tendo como base a biblioteca LightText (LT) de recuperação textual.

2 - Características:

- Primeira versão deve considerar, para indexação e pesquisa, somente os arquivos *RTF (Rich Text Format)*, de domínio público. Em caso de confirmação de negociação, a empresa Fácil prontificou-se em enviar o formato dos arquivos proprietários, para que seja atualizado o DocFind;
- Deve-se considerar primeiro que os módulos serão criados pelos projetistas do *software* hospedeiro. Estes módulos deverão compor a máquina de indexação de arquivos e as operações de pesquisa e recuperação textual, que formam o corpo principal do produto.
- A partir disto definiu-se que a API (*add-on*) será disponibilizada como uma DLL. Desta maneira a indexação poderá ser executada quando o usuário da API desejar. Da mesma forma, a DLL

permitirá que suas funções sejam chamadas quando da execução de algum controle da interface elaborada, a qual será desenvolvida pela empresa proprietária do *software* hospedeiro.

2.5.2 - Especificação Funcional

Esta atividade se concentra em detalhar, cada módulo do projeto, em funcionalidades específicas, dando uma conotação algorítmica à especificação. Isto viabilizará para a fase de desenvolvimento, que sejam especificados os objetos e operações do sistema.

Em ambos estudos-de-casos, as funcionalidades estão relacionadas às ações dos módulos de indexação e de pesquisa e recuperação textual. No entanto, no caso do DocFind, o projeto arquitetural visava formar uma base de conhecimento para a definição das funções pertencentes à API, de modo a proporcionar, aos técnicos da empresa Fácil, um ferramental que lhes possibilitasse implementar os módulos acima citados.

2.5.2.1 - Padronização da Interface com o Usuário

A presente seção busca estabelecer procedimentos para a criação da interface que o *software add-on* irá disponibilizar ao usuário. Vale salientar, contudo, que podem haver *add-ons* em que não existam interfaces com o usuário final da aplicação, como é o caso do DocFind. Neste caso, a análise é deixada em outro plano e se estabelecem os protótipos das funções que irão compor o *add-on*.

Assim sendo, o principal objetivo é fazer com que a interface do novo *software*, esteja em sintonia com as principais características da interface do *software* hospedeiro. Salienta-se que o produtor do hospedeiro poderá fornecer informações preciosas sobre as interfaces, minimizando assim esforços de análise.

Seguindo as tendências do mercado, não serão levadas em consideração as interfaces do tipo texto, concentrando-se a análise nas

interfaces gráficas de usuários (GUIs). Tal proposta visa analisar a interface do *software* hospedeiro a partir de um conjunto de *check-lists*, que identificarão quais funcionalidades de interface foram encontradas no mesmo.

Para a elaboração destes *check-lists*, utilizam-se duas abordagens sobre estilos de interação de *software*.

[SHNE92] explora cinco estilos primários de interação, sendo:

- Seleção de menu;
- Preenchimento de formulários;
- Linguagem de comandos;
- Linguagem natural;
- Manipulação direta;

Já a segunda abordagem está de acordo com a interface do Macintosh, tendo sido esta a primeira GUI existente. [HYBR89] diz que para uma interface ser gráfica ela necessita ter:

- Dispositivos de apontamento;
- Conjunto de menus de tela;
- Janelas de tarefas;
- Ícones representando arquivos e diretórios;
- Caixas de diálogo, botões, *check boxes*;
- Recursos de manipulação direta;

Com uma listagem específica das técnicas de criação de interfaces utilizadas, pode-se projetar a interface do *software add-on* de forma condizente com os estilos do *software* que irá agregá-lo.

Para isto, realiza-se a filtragem das técnicas utilizadas pelo *software* hospedeiro. Assim, forma-se um *check-list* bastante específico, chegando ao final desta seção com o *check-list* definitivo para o *software add-on*.

Desta forma, o restante da seção trata de definir, a cada item, as características relevantes de interface e, em paralelo, são expostos seus respectivos *check-lists*. Juntamente com cada um destes, está a análise do estudo-de-caso do *software* Word e do *add-on* DocM, constando algumas das telas no Apêndice A.3.

➢ Análise de Ações e Formas de Alcance

Realizando-se uma análise das principais funcionalidades do *software* hospedeiro, pode-se levantar a importância das mesmas, bem como suas formas de execução.

A partir da verificação da frequência de uso de algumas de suas tarefas, deve-se delimitar o "padrão" usado para a execução de cada um destes serviços. Vale salientar que não é necessário a avaliação completa do *software* hospedeiro, mas sim a verificação das ações semelhantes às que existirão no *add-on*. Um exemplo disto poderia ser uma tarefa de abertura de um arquivo, supondo-se ser uma ação de frequência intermediária e que também estaria presente no *add-on* a ser projetado.

AÇÕES	FUNCIONALIDADE	FORMA DE ALCANCE
Frequentes	Seleção de Fontes	Menu (Formatar-Fontes) ou por seleção de texto e definição por listas ou por estilos de texto (2ª. lista)
	Desfazer	Menu (Editar-Desfazer) ou Atalho (CTRL+Z) ou ícone ou lista de ações (anteriores ou posteriores)
de Frequência Intermediária	Abertura de Arquivos	Menu (Arquivo-Abrir) ou Atalho (CTRL+A) ou ícone abrir
	Fechamento de Arquivos	Menu (Arquivo-Fechar) ou ícone canto superior-esquerdo
Menos Frequentes	*Localizar	Menu (Editar-Localizar) ou Atalho (CTRL+L)
Infrequentes	Propriedades	Menu (Arquivo-Propriedades)

*Esta funcionalidade é a motivação para a criação do *add-on* DocM.

Tabela 2.1 - *Check-List* 1 - Frequência de Ações e Formas de Alcance

➢ Análise de Técnicas para Interação

Neste *check-list*, respectivo às técnicas de interação, mesclam-se as duas abordagens citadas anteriormente. Tem-se, na primeira coluna, a identificação da técnica e, a seguir, a sua situação (*status*) - se presente ou não -. Pode-se, ainda, ter uma terceira coluna para possíveis observações sobre características relevantes de tal estilo.

ESTILO / TÉCNICA	STATUS
Seleção de Menus	X
Preenchimento de Formulários	X
Linguagem Natural	
Manipulação Direta	X
Janelas de Tarefas	X
Ícones p/ Arquivos e Diretórios	X
Caixas de Diálogo	X
Botões	X
Sliders	
Check Boxes	X
Listas	X

Tabela 2.2 - *Check-List* 2 - Técnicas/Estilos de Interação

➢ Linguagem de Comandos

Deve conter, neste *check-list*, os tipos de linguagens de comandos, considerando-se sua complexidade. Serão então marcados, através da coluna correspondente ao *status*, a sua presença ou não no *software* hospedeiro.

LINGUAGEM DE COMANDOS	STATUS
Sintaxe Simples	X
Sintaxe Moderada	X
Sintaxe Complexa	

Tabela 2.3 - *Check-List* 3 - Linguagem de Comandos quanto à Sintaxe

➢ Métodos de Destaque

Selecionaram-se sete tópicos relativos a como destacar informações em uma interface de *software*. A estrutura deste *check-list* assemelha-se à anterior, diferenciando-se pelo acréscimo de uma coluna, reservada para uma breve descrição do tipo de destaque que está sendo utilizado pelo *software* hospedeiro.

MÉTODOS DE DESTAQUE	STATUS	DESCRIÇÃO / OPÇÕES
Intensidade		
Marcação		Sublinhado vermelho
Tamanho	X	Ícones com tamanho 32x32 pixels
Tamanho de Fontes	X	Fontes de Menus e Formulários do tipo Arial com tamanho 10 pontos
Vídeo Inverso		
Objetos Piscantes	X	
Áudio	X	Beeps rápidos (frequência não identificada)

Tabela 2.4 - *Check-List* 4 - Métodos de Destaque de Informação

➢ Uso de Cores

Este *check-list* visa identificar as cores constantes na interface do *software* hospedeiro. Para isso foram colocadas as 16 cores básicas, buscando realizar uma análise mais concisa.

Dividiu-se esta tabela de modo a identificar as cores principais da interface do produto. Em outra coluna estão os campos para as possíveis cores piscantes da interface, onde a principal é a cor padrão e a outra coluna identifica a cor de destaque.

		CORES PRINCIPAIS DA TELA	CORES PISCANTES	
			Principal	Destaque
	Cinza Claro	X		
	Cinza Escuro	X		
	Amarelo Escuro			
	Vermelho Escuro			
	Magenta Escuro			
	Verde Escuro			
	Ciano Escuro	X		
	Azul Escuro	*X		
	Branco	X		X
	Amarelo		X	
	Vermelho			
	Magenta			
	Verde			
	Ciano			
	Azul			
	Preto	X		

*Quando definida pelo ambiente operacional.

Tabela 2.5 - *Check-List* 5 - Uso de Cores

➢ Métodos de Ajuda

Formou-se esta seção do *check-list* tendo como motivação a idéia de que se está agregando uma nova funcionalidade ao *software* hospedeiro. Considera-se, portanto, que o público alvo deverá ter facilidades para adquirir o conhecimento necessário para bem operar os recursos do *software add-on* que está sendo agregado.

A tabela resultante é bastante simples, encontrando-se, na primeira coluna, o tipo de ajuda eletrônica que está sendo usada pelo *software* original e, na outra coluna (*status*), a indicação de sua existência ou não.

MÉTODOS DE AJUDA	*STATUS*
Manuais Online	X
Manuais Impressos	X
Ajuda Sensível ao Contexto	X
Tutoriais, Demonstrações e Animações Online	X

Tabela 2.6 - *Check-List* 6 - Métodos de Ajuda

➢ Dispositivos de Interação

Sendo o último *check-list*, busca-se identificar quais dispositivos que o *software* hospedeiro requer para que o usuário possa interagir com o aplicativo.

Para o preenchimento desta tabela, vale o cuidado para marcar a situação dos dispositivos que estão subdivididos, como o caso de leitoras de código. Aconselha-se que o preenchimento seja individual, especificando assim qual a leitora usada, se de barras ou magnética. O mesmo vale para posicionadores/selecionadores.

ENTRADA/SAÍDA	DISPOSITIVOS DE INTERAÇÃO	*STATUS*
DISPOSITIVOS DE ENTRADA		
	Teclado/Teclas Especiais	X
	Posicionadores/Selecionadores: Mouse Trackerball Joystick Caneta ótica Teclas de controle Tela sensível ao toque	X
	Leitoras de código: Barras Magnético	
	Tablete digitalizador	
	Scanner	
	Luvas/Capacetes/Macacões	
DISPOSITIVOS DE SAÍDA		
	Monitores de vídeo	X
	Áudio	
	Impressoras	X

Tabela 2.7 - *Check-List 7* - Dispositivos de Interação

➢ Requisitos para a Interface do *Add-on*

Como requisito básico para o início da especificação da interface do *software add-on*, deve-se ter especificado as funcionalidades reais do *software* a ser projetado. Compromete-se com isto por concordar com a afirmativa de [SHNE92], que decreta ser errôneo mudar-se alguma funcionalidade no *software* devido a novos, ou inviáveis, recursos de elaboração de interface.

Isto também se faz necessário para que não sejam analisadas gratuitamente funcionalidades do *software* hospedeiro, já que estas não trariam alterações no modo de se projetar a interface do novo *software*.

Sendo assim, devem-se identificar quais são as características de interface encontradas no *software* hospedeiro. Para tal, segue-se com a análise do *check-list* correspondente com aquele produto. Desta forma, o *check-list* terá a funcionalidade de servir como um guia de regras, que deverão ser seguidas, para que a interface do novo *software add-on* torne-se condizente com os estilos utilizados pelo *software* hospedeiro.

Desta forma, tem-se no *check-list* final apenas as características/estilos de interação que estavam presentes no *software* hospedeiro, estando prontas para que sejam marcadas ou não, conforme tenham sido satisfeitas pela interface do *software add-on*.

Salienta-se ainda, que a tarefa de criar este último *check-list*, pode se assemelhar com a atividade de teste e aceitação. No entanto, isto é consequência de se estar realizando o preenchimento do mesmo no instante em que está sendo definido ao leitor. Contudo, ressalta-se que esta lista deve ser criada a priori da definição da interface do *software add-on*.

TÉCNICAS / ESTILOS USADOS PELO HOSPEDEIRO	USO DO ESTILO NO *ADD-ON*
Ações (Frequentes, Frequência Intermediária, Menos Frequentes, Infrequentes)	FI; MF
Seleção de Menus	SIM
Preenchimento de Formulários	SIM
Manipulação Direta	SIM
Janelas de Tarefas	SIM
Ícones p/ Arquivos e Diretórios	SIM
Caixas de Diálogo	SIM
Botões	SIM
Check Boxes	SIM

Listas		SIM
Sintaxe Simples		NÃO
Sintaxe Moderada		SIM (c/ sintaxe complexa)
Tamanho		NÃO
Tamanho de Fontes		NÃO
Objetos Piscantes		NÃO
Áudio		NÃO
	Cinza Claro	SIM
	Cinza Escuro	SIM
	Ciano Escuro	NÃO
	Azul Escuro	SIM
	Branco	SIM
	Amarelo	NÃO
	Preto	SIM
Manual On-line		SIM
Manual Impresso		NÃO
Ajuda Sensível ao Contexto		SIM
Tutoriais, Demonstrações e Animações Online		NÃO
DISPOSITIVOS DE ENTRADA		
Teclado/Teclas Especiais		SIM
Posicionadores/Selecionadores: Mouse		SIM
DISPOSITIVOS DE SAÍDA		
Monitores de vídeo		SIM
Impressoras		NÃO

Tabela 2.8 - *Check-list* 8 - Estilos para a Interface do *Software Add-on*.

2.5.2.2 - Definição da Interface com o Usuário

Para a definição da interface com o usuário, utilizam-se dos documentos referentes à padronização da interface e aqueles relacionados com as definições dos módulos funcionais.

A partir dos módulos, poder-se-á discorrer brevemente sobre a presença ou não de interface com o usuário e suas funcionalidades. Em caso positivo, especificam-se os grupos de informações de entrada e as operações existentes em cada tela da interface. Por fim, inicia-se a construção do protótipo das telas, sempre procurando utilizar dos recursos de interface estabelecidos na Tabela 2.8.

O item de padronização das interfaces auxilia na busca de sintonia entre os produtos de *software add-on* e hospedeiro. No entanto, o produtor do *add-on* não pode garantir que o *software* hospedeiro em questão tenha níveis aceitáveis de usabilidade, podendo no máximo confiar nos resultados da pesquisa de mercado.

Portanto, cabe ao produtor do *add-on* gerar um produto que, além de suprir a falta de funcionalidade vislumbrada no hospedeiro, torne o usuário mais satisfeito com o produto original.

Assim, deve-se submeter o produto em questão aos testes de usabilidade, tendo como meta a certificação de que as pessoas estão trabalhando facilmente com ele.

Dentro do ciclo de vida proposto para *software*, [AZEV96] propõe o teste de usabilidade nas fases de concepção, desenvolvimento e preparação. No entanto, para *software add-on*, além de ser um método para identificar oportunidades, acredita-se que sua maior importância esteja junto a análise das interfaces e das funcionalidades desenvolvidas, testando-se o protótipo do *add-on* aqui na concepção e quando completado o desenvolvimento, sendo que o objetivo do *add-on* é de agregação de valor funcional. Contudo, na preparação

também serão necessários os testes de usabilidade, conforme a situação em que esteja inserido comercialmente o *add-on*.

Desta forma, para *software* do tipo *add-on*, os responsáveis em aplicar e analisar os testes de usabilidade a serem desempenhados nas próximas duas fases - desenvolvimento e preparação - deverão formar os grupos de pessoas, a testarem o produto, a partir da base existente de usuários do *software* hospedeiro.

Somente após o término desta atividade, a especificação funcional estará concluída e poderá fornecer subsídios para se estabelecerem as características operacionais de mais baixo nível do projeto, o que está melhor detalhado na primeira seção do próximo capítulo.

CAPÍTULO

3 - DESENVOLVIMENTO, PREPARAÇÃO E DISPONIBILIZAÇÃO

3.1 - Desenvolvimento

A fase de concepção se caracteriza por detalhar os aspectos de viabilidade e planejamento. No entanto, sua abrangência busca afunilar o processo até a atividade monofásica de especificação, após ter sido aprovada a produção do *add-on*.

Desta maneira a atividade de desenvolvimento, visualizada através da Figura 3.1, inicia com o projeto do *software*. Parte-se da especificação funcional buscando demarcar funções e operações, tornando possível estabelecer seus respectivos objetos.

No entanto não se utilizou, nos estudos-de-casos do *add-on*, de nenhuma metodologia de orientação a objetos. Foi levado em consideração o *expertise* da equipe e a técnica de abstração. [WASS96] afirma que, em desenvolvimento orientado a objetos, a abstração provê uma definição clara

das operações (métodos) para uma classe, bem como especifica os parâmetros necessários para instanciar uma classe de uma outra classe genérica.

A exaustão da especificação traz benefícios com a otimização do processo de codificação, minimizando esforços e agilizando o gerenciamento do produto e do pessoal de desenvolvimento. Isto aumenta a confiabilidade do produto final, o que é imprescindível a um *software add-on*, pois este deve proteger a integridade dos dados mantidos pela aplicação hospedeira.

Outro benefício trazido por esta segmentação é que a documentação gerada viabiliza o retorno, quando necessário, a pontos específicos da especificação, a fim de alterá-la ou, até mesmo, quando de uma guinada no projeto.

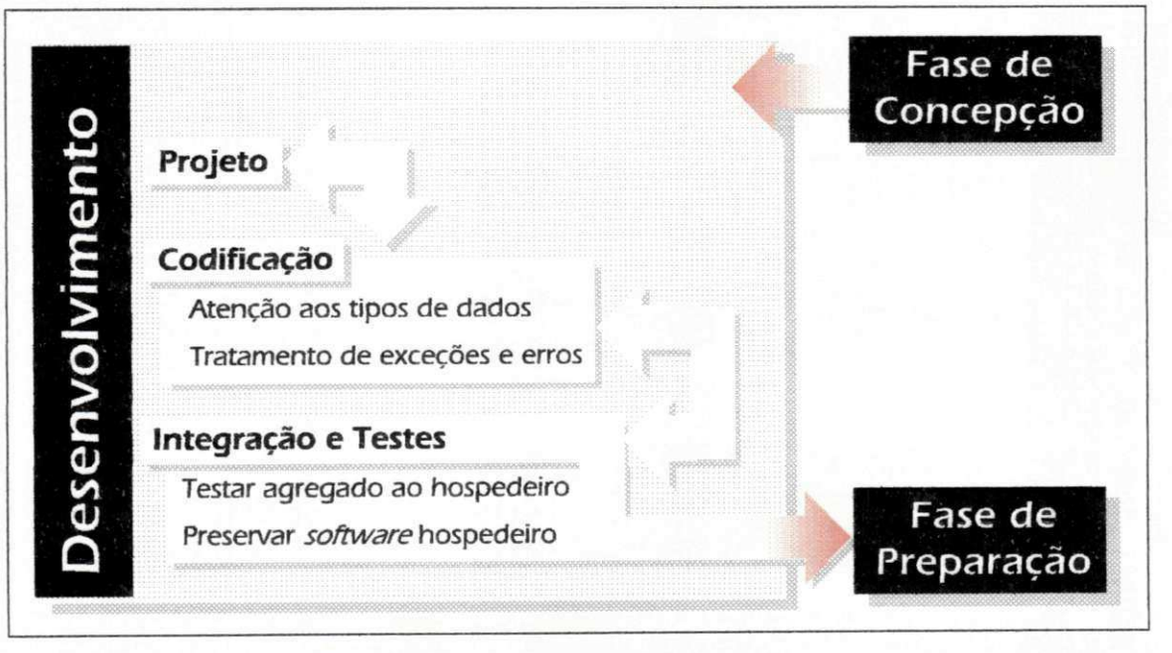


Figura 3.1 - Síntese das Atividades para a Fase de Desenvolvimento

3.1.1 - Projeto

A definição do projeto se concretiza por estabelecer as funções e operações intrínsecas aos módulos definidos na seção 2.5.

Nesta atividade, segue-se a especificação funcional a fim de dividir cada um de seus módulos em macro-operações ou, no caso de se estar desenvolvendo uma API, se dará origem às funções que a compõe.

No caso do *add-on* para o Word (DocM), foram assim definidas as macro-operações:

- Indexar;
- Desindexar;
- Alterar Nome;
- Alterar Resumo;
- Selecionar;
- Pesquisar:
    - *Control* Nome de Arquivo;
    - *Control* Texto no Documento;
    - *Control* Data de Criação;
    - *Control* Data de Atualização;
    - *Control* Resumo;
    - *Control* Botão de Pesquisa;
- Localizar Arquivos.

Para tanto, quando da mudança de projeto, pode-se retornar à especificação funcional, delimitando-a novamente e redefinindo suas funções. Assim expõe-se, na tabela abaixo, a relação das funções definidas para o *add-on* DocFind, formando sua API e sendo estas disponibilizadas aos usuários (técnicos da empresa Fácil).

Outra medida de projeto, foi estabelecer uma nomenclatura para seguir durante a implementação, buscando facilitar a manutenção e auxiliar o usuário interno, no caso os programadores que necessitam utilizar algum objeto

implementado por outra pessoa. Para isto usou-se de aspectos da terminologia húngara que, em suma, consiste em se referenciar as declarações dos tipos de dados de forma que inicie com a letra mais significativa do respectivo tipo.

Função Macro	Parâmetros	Retorno	Protótipo
Inicializar	char *: percurso destino do sistema de arquivos	int: *flag* de erro.	Int DOCF_Inicializar(char *pszPercurso);
Finalizar	nenhum	int: flag de erro.	int DOCF_Finalizar(void);
Indexar	char *: nome completo (com percurso) do arquivo	int: flag de erro.	int DOCF_Indexar(char *pszArquivo);
Desindexar	char *: nome completo (com percurso) do arquivo	int: flag de erro.	int DOCF_Desindexar(char *pszArquivo);
Alterar Nome de Arquivo	char *: nome completo (com percurso) do arquivo de origem char *: nome completo (com percurso) do arquivo de destino	int: flag de erro.	int DOCF_AlterarNome(char *pszArquivoAntigo, char *pszArquivoNovo);
Alterar Resumo do Arquivo	char *: nome completo (com percurso) do arquivo	int: flag de erro.	int DOCF_AlterarResumo(char *pszArquivo);
Pesquisar	char *: expressão de pesquisa	int: número de arquivos localizados	int DOCF_Pesquisar(char *pszExpressão);
Obter Nome de Arquivo Localizado	int: índice da lista de arquivos localizados	char *: nome completo do arquivo localizado	int DOCF_ObterNomeArquivo (int iIndice);

Tabela 3.1 - Especificação das Funções Integrantes da API DocFind.

Independentemente da abordagem, o próximo passo é definir as operações que envolvem cada uma destas funções definidas na Tabela 3.1. Seguindo-se ainda a especificação funcional, se estabelece o conjunto de operações de uma maneira algorítmica, tratando situações condicionais apenas de alto nível.

Havendo necessidade de interface com usuário, esta especificação também deve considerá-la, definindo-se o comportamento para cada um de seus componentes operacionais.

Para concluir a elaboração deste projeto, são estabelecidas as classes de objetos reais do sistema, seus métodos e atributos.

Esta atividade estando encerrada, a documentação gerada viabiliza o retorno à especificação e permite que se dê origem à elaboração do manual do usuário. No caso da API, configura-se no manual do programador, o qual é mais sintetizado e que não requer, na maioria das vezes, que seja disponibilizado de forma *on-line* no *software*. Maiores detalhes sobre manuais são apresentados nas seções seguintes.

3.1.2 - Codificação

Definira-se, ainda na especificação funcional, a plataforma de desenvolvimento e, conforme requisito do *software* hospedeiro, estabelecera-se também o método para se integrar os produtos de *software* (ver itens 2.5.1.1 e 2.5.1.2). Assim, em ambos estudos-de-casos, utilizou-se da linguagem C++ e, especialmente no caso do DocFind, a integração dar-se-ía através de uma biblioteca de ligação dinâmica (DLL - *dynamic link library*), que possibilita que aplicações desenvolvidas em Delphi/Pascal[4] (processador Fácil) executem funções implementadas em outra linguagem.

Tendo-se o projeto e as especificações sempre em mãos, o primeiro passo da codificação consiste em criar as classes e sub-classes de objetos e seus atributos. A seguir verifica-se cada operação, antes relacionada aos objetos especificados, buscando convertê-las em protótipos dos métodos públicos, pertencentes a cada classe de objeto, conforme mostra o trecho de código a seguir.

[4] Delphi é marca registrada de Borland Corporation

```
#ifndef         _DOCUMENT
#define         _DOCUMENT

#include        <stdio.h>
#include        "../include/dfconst.h"

#ifdef          __cplusplus
extern "C"
{
#endif

class   DFC_DOCUMENTO {
        private:
                FILE                    *pfDocumento;
                char                    szNomeArquivo[ MAXTAMNOME + 1 ];
                unsigned long           ulTimeCriacao;
                unsigned long           ulTimeAtualizacao;
                char                    *_ConverteData( unsigned long );
        public:
                DFC_DOCUMENTO( void );
                ~DFC_DOCUMENTO( void );

                int                     DOCF_SetarInformacao( char * );
                unsigned long           DOCF_ObterLocalTime( void );
                char                    *DOCF_ObterNomeArquivo( void );
                char                    *DOCF_ObterBasename( void );
                char                    *DOCF_ObterDataCriacao( void );
                char                    *DOCF_ObterDataAtualizacao( void );
                FILE                    *DOCF_ObterFilePoint( void );
};

#ifdef  __cplusplus
}
#endif

#endif
```

Os métodos privados e protegidos também podem ser aqui definidos. No entanto, buscando agilizar o desenvolvimento, a próxima tarefa é distribuir cada módulo do sistema entre os integrantes da(s) equipe(s), pois acredita-se que o projeto tenha sido definido de forma modular e de fácil integração a posteriori.

Quando da implementação de um *software* do tipo *add-on* - especialmente uma API - deve-se ter um cuidado especial com as definições de tipos de dados e com o tratamento de exceções e erros. No caso das estruturas de dados, deve-se focar a atenção nos tipos aceitos pela linguagem em que foi codificada a aplicação hospedeira, para não causar transtornos de integração.

Para o tratamento de exceções e erros, a saída ainda é ter uma sessão específica do manual do usuário, a qual deve relacionar os possíveis erros quando da execução de qualquer função pertencente à API.

No entanto, nem todas estas medidas precisam ser assumidas neste momento. Isto é possível porque pode-se, a qualquer instante, retornar a este ponto e implementá-las, com maior objetividade em relação às solicitações que deverão advir dos técnicos responsáveis pelo *software* hospedeiro. Segue abaixo o código pertencente a uma função da API, a qual apresenta o retorno de erros sem, contudo, identificá-lo e deixando-o para implementá-lo a posteriori.

```
int FAR PASCAL _export
DOCF_Desindexar( char *pszArquivo )
{
        DFC_REGISTRO_ARQUIVO    *pcraRegistro;
        unsigned long           ulNumeroRegistro = 0L;

        if( (pcraRegistro = G_pcca_Cadastro_de_Arquivos->DOCF_ObterRegistro(
                        pszArquivo )) == NULL ){
                return( DOCF_ERRO );
        }
        if( (ulNumeroRegistro = pcraRegistro->DOCF_ObterNumeroRegistro()) == 0L ){
                delete pcraRegistro;
                return( DOCF_ERRO );
        }
        if( G_pcsa_Sistema_de_Arquivos->DOCF_Desindexar( ulNumeroRegistro,
                        CPO_CONTEUDO | CPO_RESUMO |
                        CPO_BASENAME | CPO_DATAATUALIZACAO
                        |CPO_DATACRIACAO ) != DOCF_OK ){
                delete pcraRegistro;
                return( DOCF_ERRO );
        }
        if( G_pcca_Cadastro_de_Arquivos->DOCF_DescadastrarRegistro(
                        pcraRegistro ) != DOCF_OK ){
                delete pcraRegistro;
                return( DOCF_ERRO );
        }
        delete pcraRegistro;
        return( DOCF_OK );
}
```

3.1.3 - Integração e Testes

No momento em que foram distribuídos os módulos operacionais, entre os integrantes da equipe de desenvolvimento, já torna-se uma

consequência a interação entre os mesmos para que sejam integrados os objetos e os métodos por estes codificados.

De forma geral, quando se fala em integração de um *software add-on* ainda na fase de desenvolvimento, significa também a sua agregação ao nicho de *software* hospedeiro definido anteriormente. No entanto, poderão haver diferenças de projeto de *add-on* e, estas, trarão consigo a possibilidade de se tomarem medidas que, apesar de sutis, poderão otimizar a tarefa de integração.

Toma-se como exemplo o caso da integração e testes do DocFind. Tratando-se de uma API, e requerendo uma sequência básica de execução para suas funções, propõe-se inicialmente o estabelecimento do fluxo de execução das mesmas para que alcancem o máximo de eficiência na recuperação textual. Este fluxo também deve estar contido nos manuais do produto.

Apesar do DocFind disponibilizar um conjunto reduzido de funções, este fluxo auxilia na execução da bateria de testes e na elaboração da aplicação de demonstração. Esta, por sua vez, não foi desenvolvida na linguagem de programação a ser usada pela empresa parceira, sendo esta a nuance de projeto, antes citada, para o *add-on* em questão. Não foi necessário a geração de uma aplicação codificada em Delphi/Pascal. Isto porque o objetivo da demonstração seria de verificar o apelo do produto e sua contribuição funcional ao *software* hospedeiro.

Quanto aos testes, vale atentar especialmente ao princípio de confiabilidade, já que não se pode expor ao risco os dados mantidos pela aplicação hospedeira. No caso de uma ferramenta de indexação e recuperação textual, robustez e performance também devem receber a atenção privilegiada. Assim, cada *add-on* terá suas características específicas, porém o foco desta etapa de testes pode estar em preservar a integridade provida pela aplicação hospedeira.

Tendo-se implementada a aplicação de demonstração, pode-se estabelecer contato com a(s) empresa(s) detentora(s) do *software* hospedeiro a

fim de agendar as primeiras reuniões técnicas com objetivos também de demonstração e testes alfa e beta.

3.2 - Preparação

Após a equipe de desenvolvimento ter considerado o *software* finalizado, chega o momento em que o mesmo tem de ser transformado em um produto destinado ao mercado. Surge a fase de preparação, identificada pela Figura 3.2, que deverá tornar o produto comercializável, aplicando a este um rigoroso controle de qualidade.

Dentro do processo PDE de produção de *software*, esta fase é responsável por criar um produto de *software* tal que, ao se chegar na próxima e última fase (disponibilização), se tenha condições de vender o produto gerado, preocupando-se somente com aspectos de suporte e manutenção e com as estratégias de vendas planejadas.

As atividades a serem realizadas aqui visam garantir a qualidade e a integridade do *software*, buscando sua utilização no mercado. São desenvolvidos também todos os procedimentos de empacotamento e testes, juntamente com a documentação de aceitação dos usuários envolvidos nas atividades de testes.

No entanto, no caso de *software add-on*, o acordo de parcerias viabiliza redução de custos na fase de preparação.

Por outro lado, os casos de *add-on* considerados podem ser comercializados à parte, inicialmente, de duas formas: como um produto de prateleira por exemplo, requerendo todo o trabalho de marketing, embalagem, geração e manuais técnicos, com objetivo de conquistar os clientes do *software* hospedeiro; ou ainda como uma API, tendo seu processo implementado até a confecção de manuais, facilitando a integração entre ambas equipes de desenvolvimento (do *software* hospedeiro e do *add-on*).

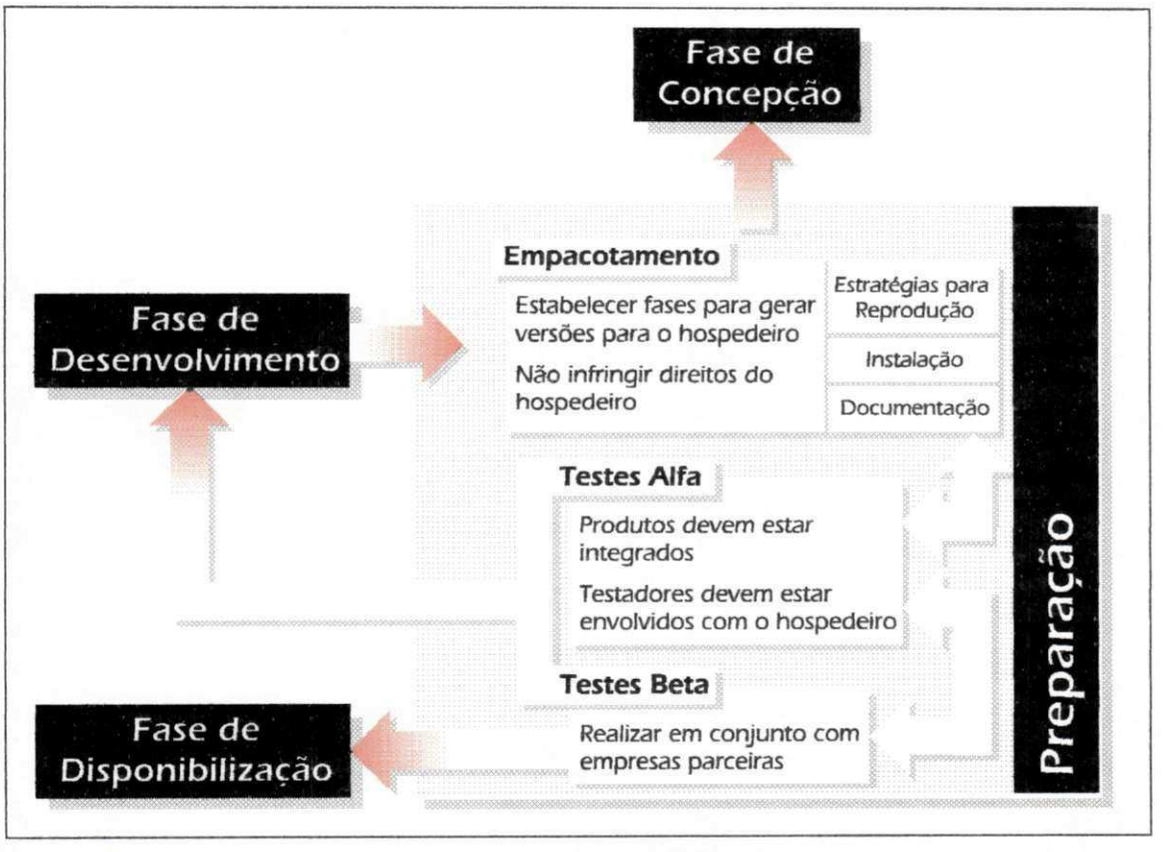


Figura 3.2 - Síntese das Atividades para a Fase de Preparação

3.2.1 - Empacotamento

Conforme a abordagem original, [MART93] trata esta atividade considerando os aspectos de instalação, proteção ao *software* e reprodução.

No caso de *software add-on*, além destes, se salientam as características tocantes ao processo de comercialização conjunta dos produtos em questão. Desta maneira, se elucidam os aspectos comerciais através dos dois primeiros itens desta seção. Já as diretrizes a serem seguidas para instalação e documentação, por serem dependentes do processo de comercialização, estão descritas nos itens técnicos que encerram esta atividade de empacotamento.

3.2.1.1 - Estratégias para Empacotamento e Reprodução

O empacotamento necessita que sejam ativados procedimentos específicos para que o conjunto de componentes, que formam o produto, esteja de acordo com o controle de qualidade e condizente com o que foi projetado e planejado para ir ao mercado.

Para *software add-on*, se o mesmo é voltado ao mercado vertical, o volume para reprodução não é significativo. Assim, a preocupação deve estar na qualidade da confecção dos manuais, mesmo que estes estejam mais voltados à equipe técnica responsável pelo *software* hospedeiro. Isto porque, estes e os outros profissionais da empresa contratante, deverão estar assistidos no momento do empacotamento de sua próxima *release* e de seus manuais.

Já no caso do segmento horizontal, a melhor alternativa é alcançada pelo acordo comercial firmado com o produtor do *software* hospedeiro. Isto pode viabilizar o empacotamento de ambos produtos em conjunto, sendo a melhor estratégia, visando minimizar os custos da fase de preparação.

Neste estágio do ciclo de produção de *software*, deve-se identificar a demanda existente a fim de atendê-la. Portanto, independentemente de se estar produzindo *software add-on*, já consegue-se verificar a necessidade do produtor estabelecer mecanismos específicos que viabilize sistematizar o processo de reprodução.

A preocupação deve estar no processo de interação com o parceiro, estabelecendo as fases em que este venha a requerer o *software add-on.* Neste caso, então, verificam-se diferentes situações em que o produtor do *add-on* terá de gerar e repassar cópias de seu *software* e, com isso, diferentes procedimentos podem ser adotados para reproduzir e empacotar o *software*:

1 - Cópia para Homologação;

2 - Cópia Demonstrativa;

3 - *Upgrade*;

4 - Licença de Uso em função do número de usuários;

5 - Cópia *Freeware*;

6 - *Site License*.

Agora, vale considerar o *add-on* como um produto horizontal, sendo comercializado independentemente do *software* hospedeiro. Neste caso, as alternativas existentes seguem as do segmento de *software* de prateleira em geral (alternativas 4, 5 e 6).

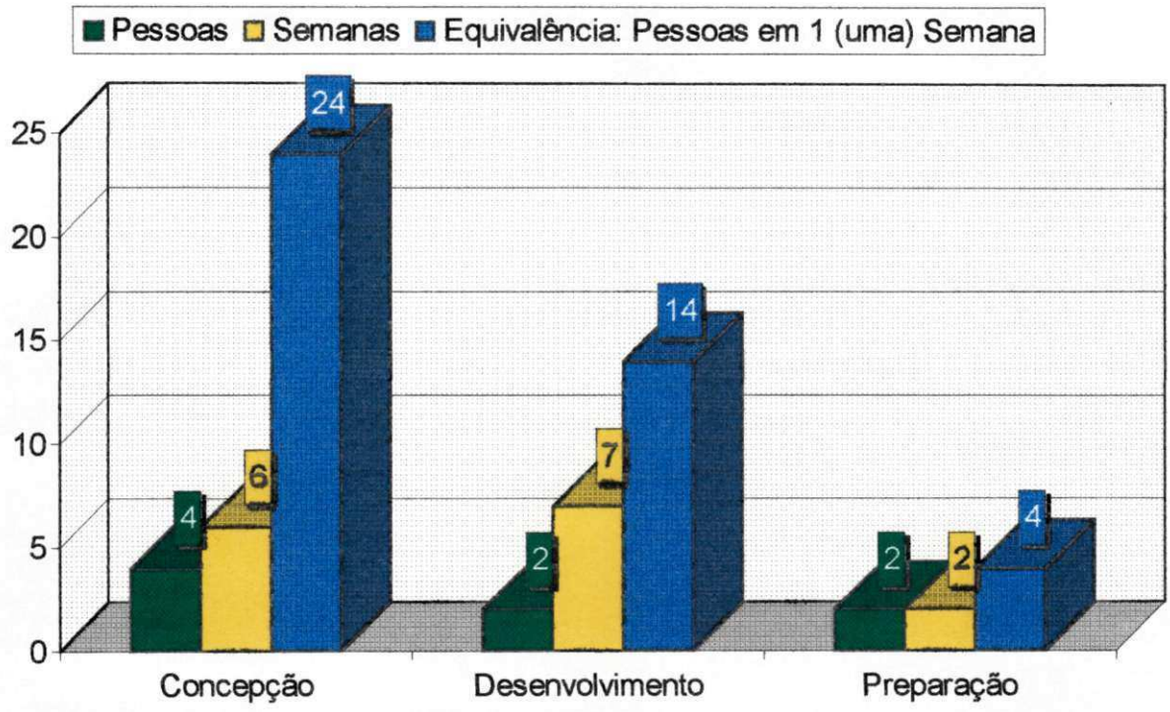
Em algumas situações, pode o *add-on* ter seus rendimentos de acordo com a estratégia de vendas adotada pelo produtor do *software* hospedeiro. Na situação 3, recebe a parcela que cabe ao *upgrade*. Já na *site licence*, a estratégia é ditada pelo departamento de marketing, que terá a meta de localizar os alvos de comercialização (ver seção 3.3).

3.2.1.2 - Controle de Qualidade

A atividade de controle de qualidade tem nesta etapa do processo uma importância acentuada, por isto este item específico do capítulo.

Considera-se que, no processo de produção de *software add-on*, o controle de qualidade deve ser aplicado mais intensamente nas fases iniciais de concepção e desenvolvimento. Isto porque, conforme o Gráfico 3.1 (apesar de não identificar o ciclo completo), identificam-se como sendo as fases mais prolongadas e, consequentemente, que requerem maiores investimentos, principalmente pelos devidos cuidados que devem ser tomados com os aspectos de integração, inerentes a qualquer projeto de *software add-on*.

Salienta-se que a relação pessoa/semana serve para identificar os esforços necessários para execução de determinada fase, nos estudos-de-casos considerados nesta dissertação.



* Não há instanciação para a fase de disponibilização

Gráfico 3.1 - Projeto DocFind: Relação Tempo-Pessoa para Fases do Ciclo de Produção de *Add-on*.

No entanto, na fase de preparação é chegado o momento de definir as estratégias de empacotamento do produto, seus aspectos legais e, principalmente, rever os acordos comerciais. Este último é que realmente viabilizará a comercialização do *add-on*, não podendo-se passar por tal processo sem documentá-lo. Além disso, o setor de qualidade tem a responsabilidade de auxiliar o setor de desenvolvimento na correção de erros, no planejamento dos testes do *software* e na especificação dos padrões internos da documentação.

Considerando a proposta do presente trabalho, o mesmo se concentra em elaborar um documento que guie os profissionais de *software* no processo de produzir um *software add-on* com menores riscos e maior controle gerencial do processo.

Desta maneira, a abordagem considerada mais flexível e adequada para o gerenciamento e evolução do processo de *software add-on*, é aquela

estendida por [SEI95]: *CMM - Capability Maturity Model*, que está descrita no primeiro capítulo, e que se origina de [HUMP88].

Já no nível 2 do CMM, as organizações produtoras de *software* devem ter seus processos, no mínimo, passíveis de serem reproduzidos em outros projetos, independentemente da equipe que compõe o mesmo. Isto somente é possível tendo-se a documentação necessária para guiar o produtor, servindo como um roteiro que balize as ações necessárias a cada fase do processo.

Não se afirma um modelo com a pretensão de alcançar os níveis mais altos de maturidade do processo para *software add-on*. Busca-se, isto sim, que a partir desta proposta seja possível se empenhar no processo de produção de tal maneira que consiga-se iniciar o processo já à frente do nível mais inferior, onde não se tem o mínimo controle sobre onde, como, e com que controle gerencial se quer chegar a um *software add-on* comercializável.

Assim, espera-se que, controlando a qualidade das atividades, nesta etapa decisiva para a comercialização, poder-se-á mais facilmente alcançar um nível de estabilidade do processo de produção.

3.2.1.3 - Procedimentos para Instalação

Sendo aparentemente uma atividade simplória, a instalação não se configura em uma tarefa isenta de erros e, além do mais, não possui um retrospecto muito satisfatório, considerando sua suposta simplicidade e sua relação com os diversos tipos de *bugs* existentes em um produto de *software*.

No entanto, o nível de complexidade das aplicações e das plataformas existentes traz consigo uma variedade de ferramentas para desenvolvimento de instalações. Estas, entretanto, cada vez mais complexas e destinadas a vários ambientes operacionais.

A criação do módulo de instalação, para *software add-on*, deve ter como premissa o *software* hospedeiro e suas respectivas instalações.

No caso de um *add-on* verticalizado, mais especificamente uma API, como o DocFind, a tarefa de gerar a instalação torna-se bastante simplificada. Além de ser uma ferramenta de desenvolvimento, a interação com os produtores do *software* original viabiliza a melhor forma de instalação, podendo esta inclusive ficar por conta do próprio *software* hospedeiro, quando da necessidade de se instalar ambos produtos de *software* em conjunto.

A última alternativa também é útil para *add-on*s horizontais, desde que a parceria estabelecida disponibilize os dois produtos em um mesmo pacote de instalação.

3.2.1.4 - Documentação

Estando presente em todas as fases do ciclo de produção de *software*, esta atividade polifásica gera um volume significativo de documentos, necessários ao controle do processo e à busca de qualidade no gerenciamento do mesmo, buscando formalizar as atividades envolvidas no processo (ver item 3.2.1.2).

No entanto, neste capítulo, é apresentado em separado, pois é na fase de preparação que devem ser concluídos todos os documentos relativos aos usuários (do *add-on* e do hospedeiro).

Como mencionado nas estratégias de empacotamento, a documentação se configura em uma peça chave na preparação do *software add-on*. Mesmo no caso de um produto horizontal, e independentemente da estratégia para comercialização, a qualidade dos manuais (de instalação, de usuário, do programador, etc.) é imprescindível para a disponibilização final do(s) produto(s) de *software*. Além disso, pode-se aproveitar a documentação do *software* hospedeiro, embutindo no mesmo os manuais referentes ao *software add-on*.

3.2.2 - Testes Alfa

Para este tipo de teste de aceitação, pode-se considerar como sendo aquele realizado internamente, no próprio ambiente de desenvolvimento. Tem-se, como precedente a este teste, o teste de desenvolvimento, o qual vem a servir como um ensaio para o teste de aceitação interno. Assim, no que tange a classificação dos testes, há sintonia com a abordagem de [MUSA96], que propõe *o Software-reliability-engineered testing - SRET*, e que pode ser de dois tipos: o teste de desenvolvimento - realizado durante a fase dois do processo - e o teste de aceitação, que engloba os testes alfa e beta.

No caso de *software add-on*, ambos os testes, alfa e beta, podem ser aplicados sob dois pontos-de-vista. Apesar disso, devem ser realizados com os dois produtos - *software add-on* e hospedeiro - já integrados. Para isto examinam-se os dois segmentos de *software add-on* até então explorados: *software* vertical e horizontal, respectivamente.

Os testes alfa a serem realizados para um segmento verticalizado devem ser mais intensos. Isto porque, com clientes em menor quantidade, certamente serão todos eles os testadores envolvidos na atividade posterior de teste beta. Assim, no caso dos testes alfa liberarem antecipadamente o produto, poder-se-á comprometer a credibilidade do mesmo junto às empresas parceiras.

Já no momento de testar um *add-on* horizontal, os procedimentos são semelhantes aos usados para testar a maioria dos produtos de *software* de prateleira. No entanto, além dos objetivos comumente visados na atividade de testes alfa, pode-se ainda buscar outros aspectos do produto realizando tais testes junto a dois grupos distintos de usuários.

Ao primeiro grupo se atribuem pessoas que dominam a operação do *software* hospedeiro, buscando seu *feedback* em relação à nova funcionalidade e a eventuais *bugs*, dentro do contexto do *software* original. Com o segundo grupo de testadores, o objeto extra de estudo será a amigabilidade oferecida

pelo novo produto e a aceitação, pelos usuários, da importância destas novas funcionalidades em relação ao *software* hospedeiro que os usuários estão conhecendo.

3.2.3 - Testes Beta

Os testes beta, quando aplicados em um segmento vertical de *software*, podem ser divididos em duas etapas distintas, conforme o público que irá compor as equipes de testadores.

A primeira etapa pode ser formada pelos usuários e técnicos pertencentes às próprias empresas contratantes do *software add-on*. Realiza-se, a posteriori, o batimento das informações oriundas de cada empresa, buscando priorizar a correção de erros e carências apontados por cada uma das empresas parceiras.

Já a segunda etapa utiliza da base de usuários estabelecida para o *software* hospedeiro. Assim, com a indicação da(s) empresa(s) parceira(s), poder-se-á disponibilizar o *software* original contendo a versão beta do *add-on*, para que o público usuário realize os testes para uma nova versão do *software* hospedeiro.

No caso de se produzir um *software add-on* horizontal, a estratégia utilizada para ativar os testes beta requer a busca dos documentos gerados na fase de concepção. Tais documentos deverão informar sobre a base de usuários do *software* escolhido durante a segmentação do mercado. A partir daí, as equipes de marketing e qualidade poderão entrar em contato com tais grupos de usuários, negociando as suas participações no processo de testes do novo *software* agregado e sobre as reais contribuições para o *software* hospedeiro.

3.3 - Disponibilização

A fase de disponibilização, dentro do processo PDE de produção de sotware, tem como meta ativar procedimentos que suportem técnica e

comercialmente o produto de *software* no mercado. Tais procedimentos se concentram em estabelecer as melhores alternativas para vender e distribuir o *software*, criando a estrutura necessária para o suporte técnico ao produto. Estas atividades são tratadas em paralelo na fase de disponibilização, conforme identificado pela Figura 3.3.

A atividade de suporte, por sua vez, poderá viabilizar, à equipe de desenvolvimento, que seja dada manutenção tanto corretiva quanto evolutiva ao produto de *software* que está sendo posto no mercado.

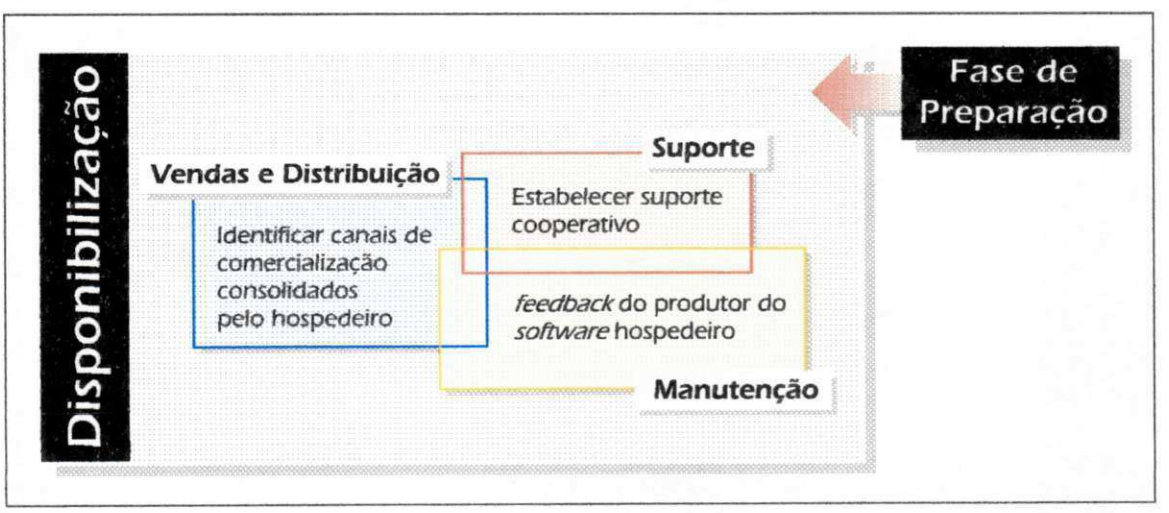


Figura 3.3 - Síntese das Atividades para a Fase de Disponibilização

Além desta fase dar continuidade ao ciclo de vida do *software*, através da detecção de novos requisitos dos usuários e do mercado, a disponibilização também é, juntamente com a preparação, a fase que requer maior fatia de tempo e volume de investimentos durante a existência dos produtos de *software* em geral.

Assim, esta fase tem importância estratégica no contexto de produção de *software add-on*. Isto porque, durante toda a fase de disponibilização, o *software* do tipo *add-on* viabiliza que sejam dadas diretrizes ao processo de tal maneira que se minimizem os recursos utilizados para se colocar e manter o *software* no mercado, pois em muitos casos será possível a

utilização de canais disponíveis para a comercialização e manutenção do produto de *software*.

Desta forma, o objetivo desta seção é identificar aspectos referentes ao comportamento do *software add-on*, quando do momento de ir para o mercado, buscando colocar em prática as estratégias para disponibilização que melhor se adequem a este tipo de *software* e que já devem ter sido estabelecidas na concepção do *add-on*.

São então abordadas as características de comercialização, suporte e manutenção dos segmentos de *software add-on* horizontal e vertical. No entanto, são enfatizados os aspectos que favorecem a disponibilização de tal produto de maneira mais cooperativa entre as empresas envolvidas, buscando-se tornar tal processo menos dispendioso ao produtor do *software add-on*.

3.3.1 - Vendas e Distribuição

A atividade polifásica de marketing, já tendo sido ativada ao longo do ciclo do *software*, é bastante importante nesta fase de disponibilização desde a concepção do *add-on*, quando dos testes de usabilidade e detecção de oportunidades e do levantamento do segmento de mercado a ser almejado.

No entanto, experiências da indústria de sofware mostram que, justamente pela grande necessidade do apelo à atividade de marketing, os custos do processo de produção de *software* podem se elevar drasticamente. Por este motivo, disponibilizar um produto de *software*, nas atuais condições de um mercado altamente competitivo, torna-se a tarefa mais árdua financeiramente dentro do ciclo de produção de *software*.

Desta forma, chega-se a um momento crucial para o *software add-on*, pois aqui há a possibilidade de serem reduzidos significativamente os investimentos necessários à disponibilização de tal produto. Poder-se-á utilizar de canais de comercialização já estabelecidos e disponíveis, quando de uma interação com as empresas parceiras, mais especificamente no caso de um produto *add-on* vertical.

Discorrem-se, ao longo desta seção, sobre as diretrizes a serem seguidas para se vender e distribuir um produto *add-on* com menores custos. Busca-se levantar as alternativas existentes para comercialização, constando de dados reais presentes no estudo-de-caso dos produtos Fácil-DocFind. Descreve-se também como alcançar os canais de comercialização já estabelecidos, utilizando-se da atividade de marketing de maneira menos dispendiosa ao produtor do *software add-on*.

3.3.1.1 - Usando Canais de Comercialização Disponíveis

O estabelecimento dos canais de venda e distribuição de *software* costumam consumir grande volume de recursos e um planejamento detalhado de marketing. Os objetivos se constituem em definir os pontos ótimos a serem atacados, a partir do mercado alvo previamente estabelecido, e as estratégias para divulgação do produto de *software*.

No caso do *software add-on*, sua característica primordial é a dependência para com outro produto de *software*. Este, por sua vez, deverá estar em uma situação mercadológica constituída. Isto irá proporcionar ao *add-on* um ciclo de vida esticado, o que se configura em uma das principais metas buscadas atualmente no âmbito da engenharia de *software*. Por conseguinte, a empresa produtora do *software* hospedeiro já deve dispor de seus próprios canais de comercialização para seu(s) produto(s).

Assim sendo, resta ao produtor do *software add-on* utilizar dos meios de distribuição estabelecidos para o nicho de mercado existente, que servirão como alicerce para a disponibilização do *add-on*. No entanto, conforme o mercado em que se estará atuando, as estratégias de busca destes canais poderão ser diferenciadas. É o caso do *software add-on* incorporado ao hospedeiro desde o seu empacotamento, ou na situação do *add-on* que se caracteriza em um produto horizontal e que necessita ir à prateleira para que seja distribuído.

Quando o *software add-on* se constitui em um produto verticalizado, a primeira medida a ser tomada foca-se em analisar a base instalada, informações estas que já devem ter sido levantadas quando do estabelecimento do nicho de *software* hospedeiro a ser atacado. O objetivo está em identificar as perspectivas de venda para o *add-on*. Assim, a Tabela 3.2 expõe a estrutura de vendas e distribuição estabelecida pela empresa produtora do processador de textos Fácil. Estas informações também são úteis às atividades de suporte e, consequentemente, de manutenção, conforme será detalhado nas últimas seções deste capítulo. Além disso, verifica-se a falta de revendas na região de origem do produtor do *add-on*. Abre-se assim a possibilidade de outra parceria no sentido de conquistar outros mercados para o *software* hospedeiro.

Estrutura	U.F.	Quantidade
Centros Fácil: distribuição, suporte e treinamento em 18 estados brasileiros	AL	1
	AM	1
	BA	2
	CE	2
	DF	3
	ES	1
	GO	3
	MG	3
	MT	1
	PA	1
	PE	1
	PI	2
	PR	6
	RJ	2
	RN	1
	RS	1
	SC	22
	SP	9
	TOTAL	62
Distribuição e Suporte via Internet	Matriz (Blumenau / SC)	1

Tabela 3.2 - Estrutura de Vendas e Distribuição da Empresa Fácil Informática.

É necessário que a equipe de planejamento e marketing já tenha verificado, durante a concepção do projeto, que a estrutura de comercialização existente irá satisfazer a meta de vendas buscada pelo *software add-on*. Caso contrário, deve-se planejar os investimentos em virtude de ações conjuntas de

atuação no mercado, o que também minimizam os custos para o produtor do *software add-on*.

Assim, o próximo passo é estabelecer um acordo para a integração comercial, especificando como se dará a disponibilização cooperativa. Este ponto envolve as alternativas de disponibilização de *software*, sendo abordado adiante.

No caso dos produtos de *software* horizontais em geral, há a necessidade de se despenderem maiores esforços para serem estabelecidos os canais de comercialização. As equipes de planejamento e marketing devem elaborar com cautela as estratégias para atuação no mercado, pois os custos envolvidos podem ser elevados.

Considerando-se mais especificamente os produtos horizontais do tipo *add-on*, verificam-se que estes também situam-se em tal cenário. No entanto, alguma estrutura já deve estar montada. Assim, a estratégia está em localizar os pontos de comercialização do *software* hospedeiro, planejando como chegar até estes e como estabelecer a ligação entre o produtor do *software add-on* e o produtor do *software* hospedeiro e seus distribuidores, revendedores, etc.

Vale salientar que, mesmo dispondo da estrutura previamente montada pelos produtores do *software* hospedeiro, o empreendedor de um *software add-on* horizontal deverá estar melhor amparado financeiramente. Isto porque será necessário o empacotamento completo do produto, o que acarreta novos custos. Além disso, a equipe de marketing deverá atuar para gerar demanda e para divulgar o *add-on*, seja independentemente, através de vendas diretas, ou por via de revendedores.

3.3.1.2 - Comércio e Marketing na Internet

Configurando-se no canal emergente para realização de negócios, a Internet oferece atualmente à indústria de *software* a melhor alternativa para expansão de mercados, tornando-se um modelo de negócios dinâmico e com

extrema redução dos custos, desde que a entrada à mesma seja implantada de forma correta e sistematizada.

Para *software add-on*, a maior utilidade ainda será para o segmento horizontal ou, no outro caso, tendo como foco empresas que buscam soluções terceirizadas, mas que trazem pouco volume de vendas e distribuição ao produtor do *add-on*.

Como ferramenta de divulgação e marketing para *software*, pode-se considerar a distribuição de "*demos*" dos produtos ofertados. É uma boa estratégia quando se quer dar aos clientes a chance de verem e se interessarem no uso do *software add-on*, podendo ser adquirido através da própria *home page* do produtor do *add-on* ou do hospedeiro. Isto economiza tempo e dinheiro em relação à forma tradicional de se oferecer versões demonstrativas de *software*, onde geralmente o mesmo é empacotado, impresso e enviado por correio aos eventuais clientes.

Assim sendo, a Internet tem uma importância fundamental para a indústria de *software*, e mais ainda ao produtor de *software add-on*, que poderá se tornar mais competitivo. Isto porque os empreendedores de *software* do tipo *add-on* poderão ser de pequeno e médio porte e mesmo assim terão a oportunidade de ingressarem no mercado internacional, bastando utilizar-se da tecnologia da rede.

> "Pequenas companhias podem agora ter uma presença global através de suas *home pages* - WWW torna a distância irrelevante -. Para muitas companhias, penetrar no mercado internacional, antes do advento da Internet, seria impossível. Muitos empreendimentos vêem a Web como uma maneira relativamente de baixo risco para penetrar em um vasto mercado e que, de outra forma, estaria fora de questão" [LESN95].

3.3.1.3 - Alternativas de Disponibilização

Para se estabelecer quais as possibilidades que o cliente terá para adquirir o *software add-on*, deve-se antes verificar quais as alternativas que o produtor do *software* hospedeiro dispõe para que o público alvo adquira o seu produto.

No caso do DocFind, verificou-se que a base instalada do processador de textos Fácil está em torno das 120 mil cópias, tendo 1500 cópias vendidas em média mensalmente. Assim, juntando-se estes dados, com a estrutura exposta na Tabela 3.2, acredita-se poder seguir as mesmas alternativas de vendas do *software* hospedeiro.

A partir daí, o próximo passo é identificar como se dará o processo de repasse dos faturamentos, o que no caso de *software add-on* vertical a solução a adotar pode ser a atribuição de *royalties* para cada cópia vendida.

Para *add-on* horizontal, as opções de comercialização se assemelham a qualquer *software* de prateleira. Ressalta-se o cuidado à estipulação do preço do produto, tendo-se cautela em relação ao aspecto custo-benefício, pois quem detém um *software* de maior porte já pagou um preço alto e, se as funcionalidades ora propostas pelo *software add-on* não forem suficientemente vantajosas, seu preço pode inviabilizar o *add-on* de forma mais intensa do que para os demais produtos de *software*.

3.3.2 - Suporte

A definição atual, para a equipe de suporte técnico, somente pode compartilhar dos aspectos até então utilizados para esta atividade no que diz respeito à sua característica primordial, que é de estar em contato direto com o usuário do produto que se deseja suportar.

Portanto, a função do setor de suporte não mais se restringe a receber reclamações técnicas de operação do produto de *software* e em seguida retornar com soluções provisórias.

O cenário é mais abrangente, onde atribui-se um papel extremamente estratégico ao pessoal envolvido com o suporte técnico, expandindo sua existência de forma a buscar a satisfação plena do usuário, oferecendo um serviço especial ao cliente e criando um importante diferencial tanto para o produto como para o fabricante do *software*.

Na implantação do suporte, os maiores gastos são efetuados em equipamentos, instalações, *software* e material de escritório. No entanto, uma vez implantada a estrutura da central de suporte, os principais custos passarão a ser os salários do pessoal de suporte [SANT95].

Conforme [SOFT93], os custos com a mão-de-obra técnica podem contabilizar, em grandes centrais de atendimento, mais de 50% dos custos de suporte. Já para pequenas centrais, os mesmos podem representar de 75% a 85% dos custos totais.

Ainda segundo pesquisas realizadas por [SOFT90] e [SOFT93], foi constatada que a mediana dos custos do suporte técnico, para as empresas americanas, era de 6% do faturamento total dos mesmos. No entanto, as taxas variam consideravelmente conforme a categoria das aplicações de *software*. Enquanto a categoria que o *add-on* está inserido costuma requerer de 3% - 4% do faturamento total, os demais produtos exigem custos na faixa de 10%.

Fica evidente que os custos envolvidos na atividade de suporte ajudam a complicar a gerência dos recursos disponíveis. Pode-se verificar, no entanto, que o *software* do tipo *add-on* contribui para que sejam minimizados os custos para operação do suporte.

Em função do segmento de mercado do *add-on*, a redução é significativa, podendo chegar a 5,7%. A atividade de suporte para *software add-on* vertical pode, como na atividade de vendas e distribuição, utilizar-se de um canal bem estabelecido para que seja desempenhada. Isto, é claro, irá depender do acordo firmado entre ambas as partes.

Existe a situação em que o produtor do *software* hospedeiro deseja se prevenir de erros advindos da ferramenta agregada. No entanto, isto pode ser evitado se a atividade de testes de aceitação deixar explícita a boa qualidade do *software add-on* e das funcionalidades por ele prestadas.

No entanto pode haver confiabilidade da empresa detentora do *software* original em relação a "suportar ambos os produtos". Na realidade o suporte aparenta ser a um único produto. Contudo, o produtor do *software add-on* estará de prontidão para receber o *feedback*, porém este advirá das equipes técnicas do fabricante do *software* hospedeiro.

Esta estratégia reduz os investimentos do produtor do *add-on* para suportar o *software*. Além disso, garante-se que as solicitações estejam mais consolidadas, pois virão de equipes que também lidam com produção de *software*. Todavia depende-se de como este desempenha a atividade de suporte, apesar de que se esta não for adequada, ambos os produtos terão seu ciclo de vida reduzidos na mesma proporção.

Agora, se for considerado o segmento de *software add-on* para comercialização independente, as características de suporte sofrem diferenças substanciais. Torna-se necessária uma estrutura de suporte com procedimentos automatizados[5] ,semelhante a dos produtos de *software* de prateleira em geral, o que aumenta drasticamente os custos e a complexidade de execução da tarefa.

Em ambas as situações, a atividade de suporte pode seguir as tendências atuais, sendo operacionalizada através da Internet. Enquanto que a coleta de dados para o suporte busca agilizar o processo de resposta ao usuário, a Internet pode prover que as dúvidas mais comumente apresentadas estejam respondidas já no *site* do fabricante do *software*. Além disso, a clientela pode dispor de informações mais atualizadas sobre o *software* de interesse,

[5] [MART93] trata com maiores detalhes sobre a automação da atividade de suporte.

onde são expostas as diretrizes de evolução do produto e as prioridades que estão sendo dadas a cada exigência do usuário. Tudo isto aumenta a credibilidade do *software* e da empresa que o produz.

3.3.3 - Manutenção

Considera-se a manutenção de *software* como sendo um conjunto de atividades da engenharia de *software* que são desempenhadas após o mesmo ter sido liberado ao cliente e colocado em operação [PRES87]. No entanto, para tornar mais condizente com a realidade da atividade, deve-se ter em mente as categorias de manutenção de *software*, criadas por Swanson [SWAN76] e largamente utilizadas pela literatura da área, sendo elas: corretiva, *adaptativa* e *perfeitiva*.

Assim, não se pode atribuir à atividade de manutenção de um *software* única e exclusivamente a correção de *bugs* (erros residuais) detectados na operação do mesmo. Devido a mudanças tecnológicas nas plataformas e sistemas operacionais, os produtos de *software* anteriormente projetados devem ser readaptados aos novos ambientes (manutenção *adaptativa*). Além disso, o mercado está em constante evolução, sugerindo avanços nos produtos utilizados e requerendo novas funcionalidades, fazendo com que os requisitos prioritários sejam adicionados ao *software* (manutenção *perfeitiva*).

Independentemente de ser um *add-on* vertical ou horizontal, o que ativará o processo de manutenção do produto terá origem no suporte e no produtor do software hospedeiro, passando pelo pessoal de gerência e marketing, os quais devem priorizar as mudanças a serem realizadas de acordo com a metodologia da empresa.

Além das categorias de manutenção supra citadas, a literatura identifica uma quarta categoria para manutenção, mormente utilizada no âmbito da tecnologia de fabricação de *hardware*, chamada de manutenção preventiva. Esta se caracteriza na manutenção que tem como objetivo melhorar a

manutenabilidade e confiabilidade do software[6], ou para prover uma base melhor para futuras evoluções no produto de *software* [PRES87].

As três primeiras categorias são amplamente difundidas na literatura técnica para que se preste manutenção a produtos de *software* em geral. No entanto, suas características não fogem muito à realidade dos demais tipos de *software*.

Assim sendo, o foco desta seção paira sobre o processo preventivo de manutenção e na gerência de configuração de *software*, pois no caso de *software add-on* o desenvolvimento ou, neste caso, a manutenção preventiva, pode se configurar em um ponto chave para a conquista de novos mercados.

Após o *add-on* ter sido disponibilizado ao mercado, além do *feedback* que o suporte obtém a respeito de erros e aperfeiçoamentos do produto, pode-se também, juntamente com a equipe de marketing, vislumbrar o respaldo da aplicação junto aos produtos hospedeiros concorrentes daquele que está agregando o *software add-on*.

Por não ser aconselhável se estabelecer exclusividade no contrato firmado com a empresa parceira, existe a viabilidade de se abrirem novos negócios junto a outros produtos de *software*. Assim, poder-se-á contactar com os produtores destes a fim de se negociar novas parcerias. Em paralelo a isso, poderão estar sendo efetuadas as devidas modificações no *software add-on*, de forma que este possa se agregar a outros hospedeiros.

Para que esta alternativa seja passível de execução, todo o processo deve ser preventivo e, principalmente, a manutenção, onde deverão ser armazenados todos os documentos inerentes a cada versão correspondente ao *software* hospedeiro envolvido. Portanto, a gerência de configuração de *software*, além de dividir o produto por versões e *releases*, deve também

[6] Manutenabilidade e confiabilidade são propriedades da engenharia de *software* e são definidas na norma ISO-9126 [ISO9126] para qualidade de *software*.

organizar todo o material do *add-on* de acordo com o produto que o está agregando.

Poder-se-ía considerar estas diretrizes como sendo inerentes ao processo de portabilidade ou a alguma das três categorias de manutenção definidas anteriormente. No entanto, o *add-on* somente pode trazer vantagens competitivas quando seu processo é tratado a rigor e, neste caso, acredita-se que o desenvolvimento e a manutenção preventiva podem trazer um diferencial para o sucesso do *software add-on*.

CAPÍTULO

4 - CONCLUSÃO E TRABALHOS FUTUROS

Considerando a competitividade e especialidade do mercado, verifica-se a constante busca por espaços ainda não explorados para comercialização de *software*.

Foi identificado que os modelos de processos de produção de *software* existentes não consideravam as especificidades de cada segmento. Assim, [MART93] propôs o molde R-Cycle para o processo PDE de produção de *software*, permitindo que o mesmo fosse instanciado e calibrado de acordo com as necessidades de cada empreendedor e do nicho de mercado por este visado.

Desta forma, o presente trabalho tratou de abordar o segmento de mercado de *software add-on*, no qual acredita-se que a indústria nacional e os pequenos empreendimentos poderão ter maiores chances competitivas com menores riscos. Porém, esta oportunidade pode ainda não estar sendo bem aproveitada pela indústria nacional - podendo ser inclusive pela falta de metodologias específicas - onde identificam-see apenas 6,5% das empresas

dedicando-se à produção de *software* embutido, conforme pesquisa realizada por [SPIA95].

O modelo proposto não tem como pretensão alcançar os níveis mais altos de maturidade do processo para produção de *software add-on*. Busca-se, isto sim, antecipar dificuldades, identificando a concepção como a fase estratégica para o add-on e que requer maiores esforços. Espera-se que esta proposta possibilite o empenho no processo de produção de add-on de tal maneira que se consiga iniciar o processo já a frente do nível mais inferior, onde não se tem o mínimo controle sobre onde, como, e com que controle gerencial se quer chegar a um *software add-on* comercializável.

Portanto, chega-se à conclusão deste trabalho com o principal objetivo de ter gerado um documento que guie os profissionais de software no processo de produzir um software add-on com menores riscos e maior controle gerencial do processo. A globalidade do modelo é melhor expressada pela Figura 4.1 a seguir, que compila todas as fases e atividades até aqui descritas para se produzir *software* do tipo *add-on*.

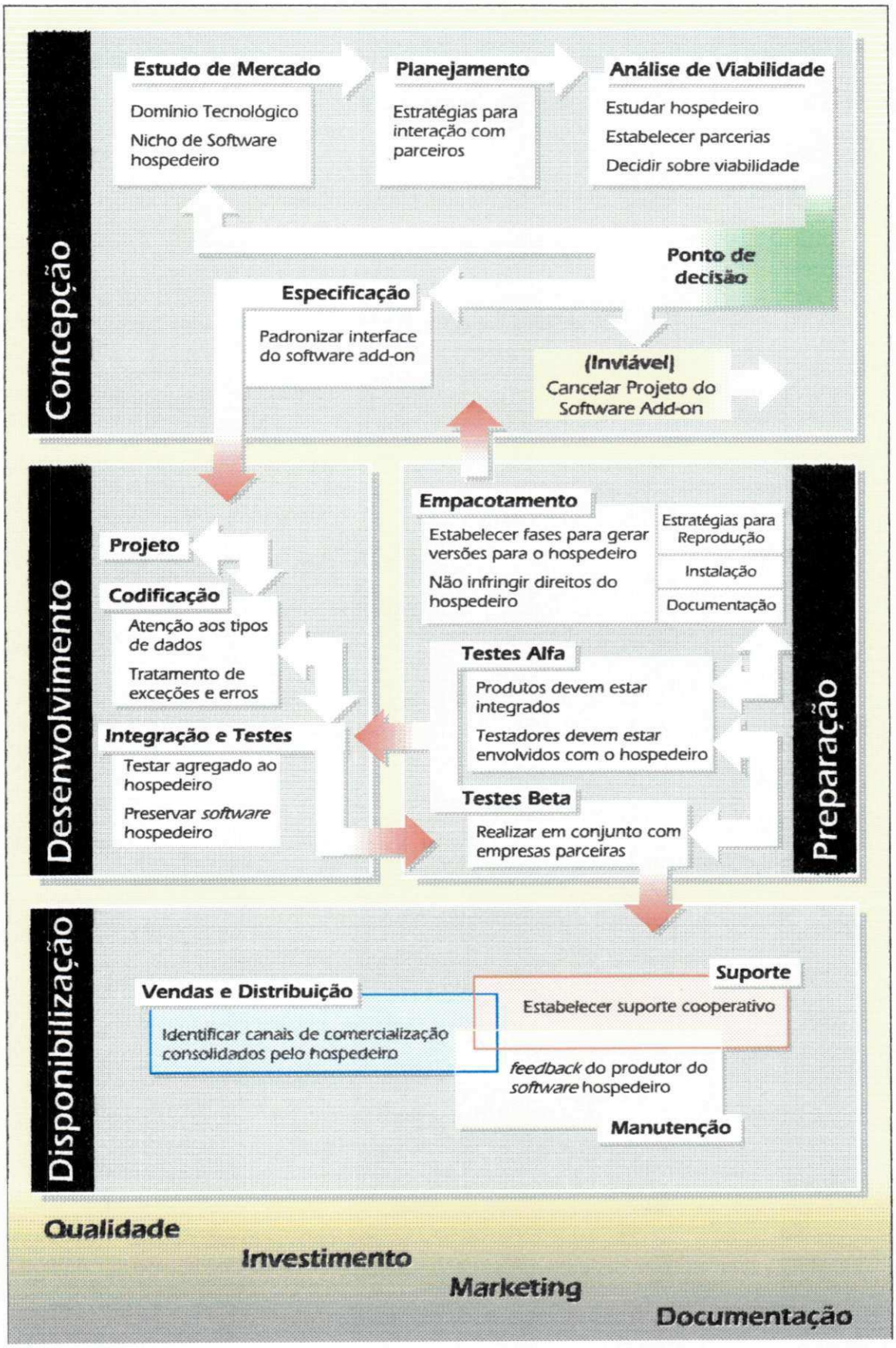


Figura 4.1 - Instanciação do Molde R-Cycle para o Processo PDE de *Software Add-on*

4.1 - Considerações Finais e Trabalhos Futuros

Acredita-se que os objetivos estabelecidos no início do trabalho tenham sido alcançados, antecipando as dificuldades existentes ao se empreender a produção de *software add-on*. O trabalho está composto de um resumo executivo e da dissertação propriamente dita, onde ambas podem ser lidas independentemente mas que, no entanto, se complementam.

O resumo executivo traz consigo informações em forma de guia prático para o setor industrial de *software* e para estudantes de informática com intenção de empreender a produção de *software add-on*. Já a dissertação forma o documento detalhado do processo de produção de *add-on*, permitindo que sejam tiradas eventuais dúvidas por aqueles que abordarem somente o resumo executivo e para que seja criticado e evoluído o modelo para *software add-on*.

Desta forma, para que o modelo proposto torne-se sempre mais abrangente e confiável, é necessária a obtenção de novos resultados a partir de outras validações do modelo.

Esperando que o presente trabalho venha a colaborar satisfatoriamente aos segmentos interessados, acredita-se que somente a cooperação entre academia e indústria poderá viabilizar testes de campo mais completos e aperfeiçoados.

Portanto, em prol de tornar o modelo enxuto e cada vez mais confiável, algumas sugestões de novos trabalhos podem ser consideradas, sendo:

- Empreender estudos-de-casos buscando abranger, cada vez mais, atividades eventualmente não abordadas neste trabalho, o que é possível, considerando-se os segmentos horizontal e vertical de *software add-on*. Cada um destes tem as suas próprias especificidades no processo de produção de *add-on*, como por exemplo a forma de comercialização e as alternativas para acordos com empresas parceiras;

- Identificar outros tipos de *software* embutidos que poderão, por suas semelhanças ao *add-on*, terem seus modelos mais facilmente criados;
- Junto aos programas SoftEX-2000, R-Cycle e Genesis, poder-se-á ajustar o modelo de acordo com a realidade da indústria nacional, buscando também novos dados relativos a produção de *software add-on* para o mercado mundial.

Referências Bibliográficas

[AZEV96] **AZEVEDO, Fernanda Angélica Tenório B.** Teste de Usabilidade de Software: Sua Aplicação nos Processos de Concepção, Desenvolvimento e Preparação. Dissertação de Mestrado, Universidade Federal da Paraíba, Dezembro, 1996.

[BANN91] **BANNERT, J.** New SEI Maturity Model Target Key Practices. IEEE Software, November, 1991.

[BUSS95] **BUSSMANN, José E. C.** BART - Uma Biblioteca Orientada a Objetos de Apoio à Recuperação Textual. Dissertação de Mestrado, Universidade Federal da Paraíba, Dezembro, 1995.

[GENE97] **GENESIS, Projeto.** http://www.dsc.ufpb.br/~Genesis, 1997.

[GHEZ91] **GHEZZI, Carlo & JAZAYERI, Mehdi & MANDRIOLI, Dino.** Fundamentals of Software Engineering. Prentice-Hall Internacional, 1991.

[HUMP88] **HUMPHREY, Watts S.** Characterizing the Software Process: A Maturity Framework in IEEE Software, pp. 73-79, March, 1988.

[HUMP95] **HUMPHREY, Watts S.** A Discipline for Software Engineering. Addison-Wesley, 1995.

[HYBR89] **HAYES, Frank & BARAM, Nick.** A Guide to GUIs. Byte, julho 1989, vol 14, Nº 7, págs.: 250-257

[ISO9126] **ISSO/IEC 9126: 1991 (E).** Information Technology-Software Product Evaluation-Quality Characteristics and Guidelines for their Use. ISO/IEC, Genevé, Switzerland, 1991.

[JONE94] **JONES, Caper.** Assessment and Control of Software Risks. Yourdon Press, 1994.

[LESN95] **LESNICK, Leslie & DAHL, Andrew.** Internet Commerce. New Riders Publishing, 1995.

[MART93] **MARTINS, Luiz M. F.** Um Molde para o Processo de Produção e Disponibilização de Software Comercial. Ilustrações com Software Unix de Prateleira. Dissertação de Mestrado, Universidade Federal da Paraíba, Dezembro, 1993.

[MART95] **MARTINS, Luiz M. F. & MOURA, José Antão B. & MEDEIROS, Álvaro F. C.** R-Cycle: Um Molde para o Processo de Produção, Disponibilização e Evolução de Software. IX SBES - Simpósio Brasileiro de Engenharia de Software, Recife, Outubro, 1995.

[MELO97] **MELO, Paulo R. S. & BRANCO, Carlos E. C.** Setor de Software: Diagnóstico e Proposta de Ação para o BNDS. http://www.softex.com.br, 1997.

[MOUR94] **MOURA, José Antão B. & MARTINS, Luiz M. F. & MEDEIROS, Álvaro F. C.** R-Cycle: Um Molde para o Processo de Produção, Disponibilização e Evolução de Software. PROTEM, 1994.

[MOUR97] **MOURA, José Antão B. & MARTINS, Luiz M. F. & MEDEIROS, Álvaro F. C.** R-Cycle: Um Molde para o Processo de Produção, Disponibilização e Evolução de Software. PROTEM, 1997.

[MUSA96] **MUSA, John D.** Software-Reliability-Engineered Testing in Computer: innovative technology for computer professionals, Vol. 29, No. 11, pp. 61-68, November. 1996.

[PRES87] **PRESSMAN, Roger S.** Software Engineering: A Practitioner's Approach, Second Edition, McGraw-Hill, 1987.

[SANT95] **SANTOS, Celso J.** Uma Proposta de Organização para o Suporte Técnico. Dissertação de Mestrado, Universidade Federal da Paraíba, Setembro, 1995.

[SEI95] **CMU/SEI, Carnegie Mellon University / Software Engineering Institute.** The Capability Maturity Model: Guidelines for Improving the Software Process. Addison-Wesley, 1995.

[SELE96] **SELECT, Computer.** Full Text COPYRIGHT 1996 The Computer Language Co. Inc. December, 1996.

[SHNE92] **SHNEIDERMAN, Ben.** Designing the User Interface-Strategies for Effective Human Computer Interaction. Addison Wesley Publishing Company,1992.

[SOFT90] **SOFT-LETTER.** Watertown: Jeffrey Tarter, Volume 7, Número 16, Fevereiro, 1990.

[SOFT93] **SOFT-LETTER.** Watertown: Jeffrey Tarter, Volume 10, Número 10, Setembro, 1993.

[SPIA95] **MINISTÉRIO DA CIÊNCIA E TECNOLOGIA, Secretaria de Política de Informática e Automação.** Qualidade no Setor de Software Brasileiro. Brasília, 1995.

[SWAN76] **SWANSON E. B.** The Dimensions of Maintenance. Proc. 2nd Intl. Conf. Software Engineering, IEEE, October 1976, pp. 492-497.

[WASS96] **WASSERMAN, Anthony I.** Toward a Discipline of Software Engineering. in IEEE Software, Vol. 13, No. 6, pp. 23-31, November. 1996.

[WEBE97] **WEBE, Kival C. & ROCHA, Ana R. C. & DE LUCA, José C. M.** Qualidade e Produtividade em Software: termo de referência do subprograma setorial da qualidade e produtividade em software, do Programa Brasileiro da Qualidade e Produtividade - PBQP. 2ª ed. Rev. E ampl., São Paulo: Makron Books, 1997.

Apêndice A

A.1 - Questionário Utilizado na Pesquisa de Mercado

Prezado usuário de Micro informática,

Solicitamos um momento de sua atenção para o preenchimento deste questionário. Suas informações serão de extrema importância para o desenvolvimento de um estudo sobre "análise de viabilidade de software para usuários de microinformática", desenvolvido pela Green Software, uma empresa vinculada ao programa nacional de exportação de software. Este questionário faz parte de um estudo de caso, cujos resultados serão utilizados numa dissertação de mestrado em progresso sobre o tema.

Caso se sinta em dúvida sobre alguma questão ou não se sinta a vontade sobre a mesma, não precisa respondê-la. O preenchimento consciente irá ajudar-nos a ter um maior sentimento sobre esta pesquisa que estamos desenvolvendo.

Muito gratos pelo apoio.

Green Software

Questionário

Marque com um X a(s) alternativa(s) de sua escolha dentre as opções

1 - Perfil do Usuário

1. Qual a sua atividade atual ?
   a. estudante b. funcionário público c. profissional liberal _____
   d. professor e. pesquisador f. empresário
   g. área administrativa e. área de informática

2. Você acha fácil e intuitivo o uso de microcomputadores ?
   a. Sim b. Não

3. Qual o seu uso diário de microcomputadores ?
   a. menos que 1 hora
   b. entre 1 e 2 horas
   c. entre 2 e 4 horas
   d. entre 4 e 8 horas
   e. mais que 8 horas

4. Qual o ambiente que você usa no seu micro ?
   a. Windows b. DOS c. OS/2 d. Macintosh
   e. Outros_____

5. Quanto tempo o seu micro fica ligado durante o dia ?
   a. nunca desligo
   b. ligo pela manhã e desligo à noite
   c. fica ligado apenas quando estou usando

6. Qual o tempo aproximado de convívio com microcomputadores ?
   a._____ anos b. _____ meses

2 - Sobre editores de textos

1. Você usa editores de textos
   a. Não b. Sim

2. Qual(is) editor(es) de texto você usa ?
   a. Word for Windows b. AmiPRO c. Word Perfect d. Wordstar
   e. Xywriter f. Carta Certa g. Write (windows) h. outros_____

3. Há quanto tempo você usa editores de textos ?
   a. Menos de 1 ano
   b. de 1 à 2 anos
   c. de 2 a 3 anos
   d. mais de 3 anos
   e. não me lembro

4. Você se considera um usuário _____ de editores de textos
   a. iniciante (usa recursos básicos e tem algumas dificuldades)
   b. intermediário (conhece os recursos básicos ao ponto de auxiliar usuários iniciantes)
   c. avançado (domina recursos como gráficos, tabelas, mala-direta,verificação ortográfica)
   d. expert (domina completamente o editor que usa)

3 - Sobre editores de textos para windows

1. Quantos documentos você cria em média por dia quando usa o editor de textos ?
   a. 1 b. 2 c. de 3 à 5 d. de 5 à 10 e. mais de 10

Apêndice A

2. Qual o tamanho médio dos seus textos (documentos) ?
   a. 1 página b. 2 páginas c. de 3 à 5 pàginas d. de 5 à 10 páginas
   e. mais de 10 páginas

3. Qual o tamanho do maior texto que você criou ?
   a. menos de 2 páginas b. de 2 à 5 páginas c. de 5 à 10 páginas
   d. de 10 à 20 páginas e. aproximadamente _____ páginas

obs. pode colocar o tamanho em kbytes ou Mbytes _____

4. Como você ativa o seu editor de textos ?
   a. através do ícone
   b. através do gerenciador de arquivos (file manager), clicando o nome do arquivo
   c. fazendo primeiro uma pesquisa no gerenciador de arquivos para localizar o arquivo
   d. com teclas de atalho do windows
   e. no grupo startup (iniciar) do windows
   f. a partir da linha de comando dos DOS
   g. outros_____

5. Preencha as lacunas com S para Sim e N para Não
   ( ) usa senhas para proteger seus arquivos de textos ?
   ( ) usa salvador de telas (Screen Saver) ?
   ( ) usa auto-salvamento de arquivos ?
   ( ) quando sai para tomar um cafézinho, você salva o arquivo ?
   ( ) deixa o arquivo aberto na tela por muito tempo mesmo quando não está usando ?
   ( ) você costuma abrir mais de um documento ao mesmo tempo ?
   ( ) costuma colocar palavras-chave e preencher os resumos informativos nos seus textos ?
   ( ) costuma usar a opção de localizar arquivos que o seu editor de textos oferece ?

<u>4 - Sobre organização e localização de arquivos</u>

1. Onde você coloca os seus arquivos de texto ?
   a. em diretórios diferentes
   b. num mesmo diretório pessoal
   c. deixo que o editor de textos se encarregue disto
   d. Outros _____

2. Preencha as lacunas com S para Sim e N para Não
   ( ) acha fácil a criação e manipulação de arquivos e diretórios?
   ( ) você localiza facilmente documentos (arquivos) no seu micro ?
   ( ) preferiria não ter que saber da existência de arquivos e diretórios, deixando que este trabalho ficasse a cargo do editor de textos ?
   ( ) desconheço a existência de diretórios
   ( ) preferiria apenas dizer para que serve o arquivo (ex: carta de de licitação) do que ter que criar um nome para o arquivo (exemplo : carta_lic.doc)
   ( ) acho essencial a existência de diretórios uma vez que organiza meus documentos

3. Aproximadamente qual o tempo que você gasta para localizar um arquivo antigo ?
   a. menos de 1 minuto
   b. de 1 à 3 minutos
   c. de 3 à 5 minutos
   d. de 5 à 10 minutos
   e. mais de 10 minutos
   f. peço auxílio
   g. geralmente desisto

4. Como você faz para localizá-lo ?
   a. através do nome (ou parte do nome) do arquivo no File Manager
   b. procuro nos diretórios um a um
   c. uso o gerenciador de arquivos do editor
   d. abro vários documentos no editor até localizá-lo
   e. através de palavras-chave e resumos informativos

5. Você já usou ou conhece algum software voltado para a gerência de documentos ?
   a. Sim usei _____ conheço _____
   b. Não
   c. Ouvi falar no software _____

6. Suponha que você faz parte da assessoria de Fernando Henrique para assuntos internacionais.
Seria interessante saber, por exemplo, quais as correspondências enviadas para a embaixada do Brasil em Washington, no período de janeiro à março de 1994 que falavam sobre importação de alimentos.

Enfim, seria produtivo localizar os seus documentos pesquisando através dos conteúdos dos documentos e não apenas pelo nome do arquivo.

a. Não b. Seria bom c. Extremamente Produtivo d. Talvez

Comentários que julgar pertinentes

_____

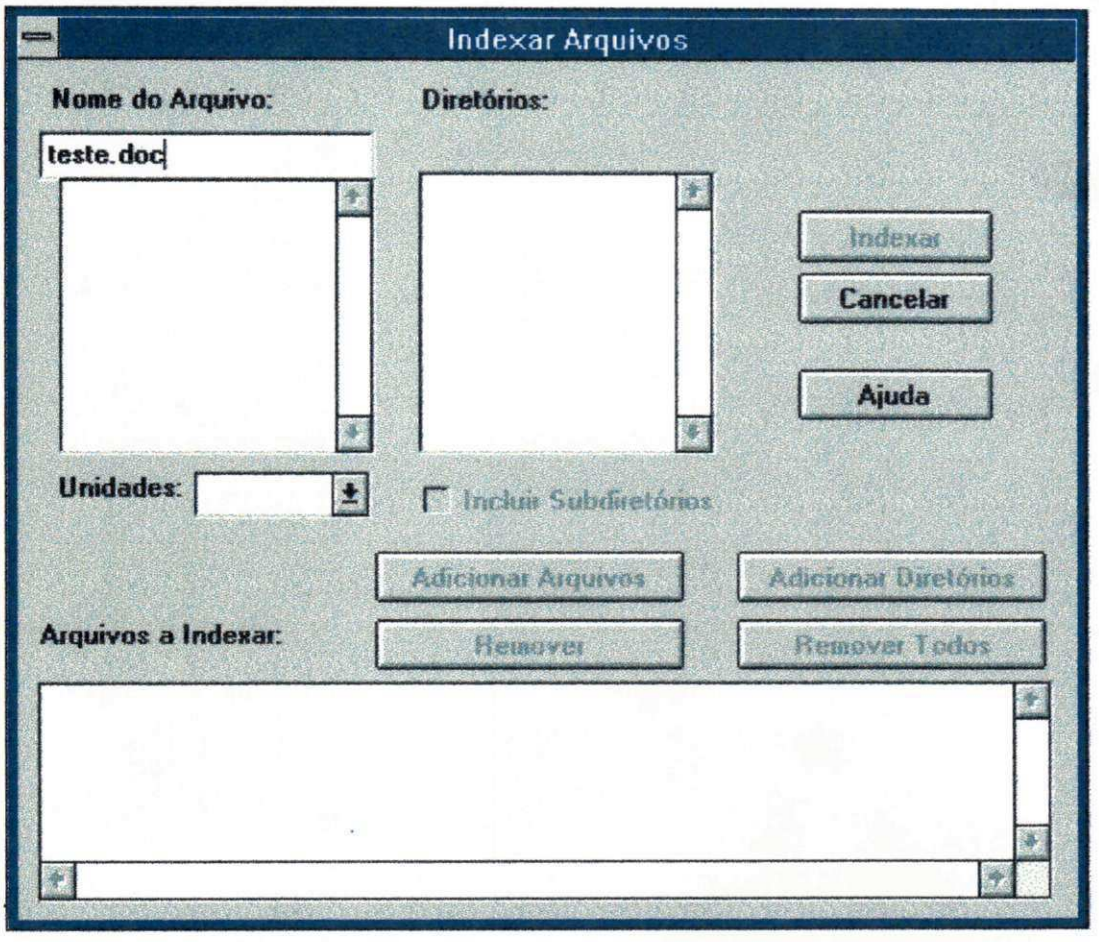
_____

_____

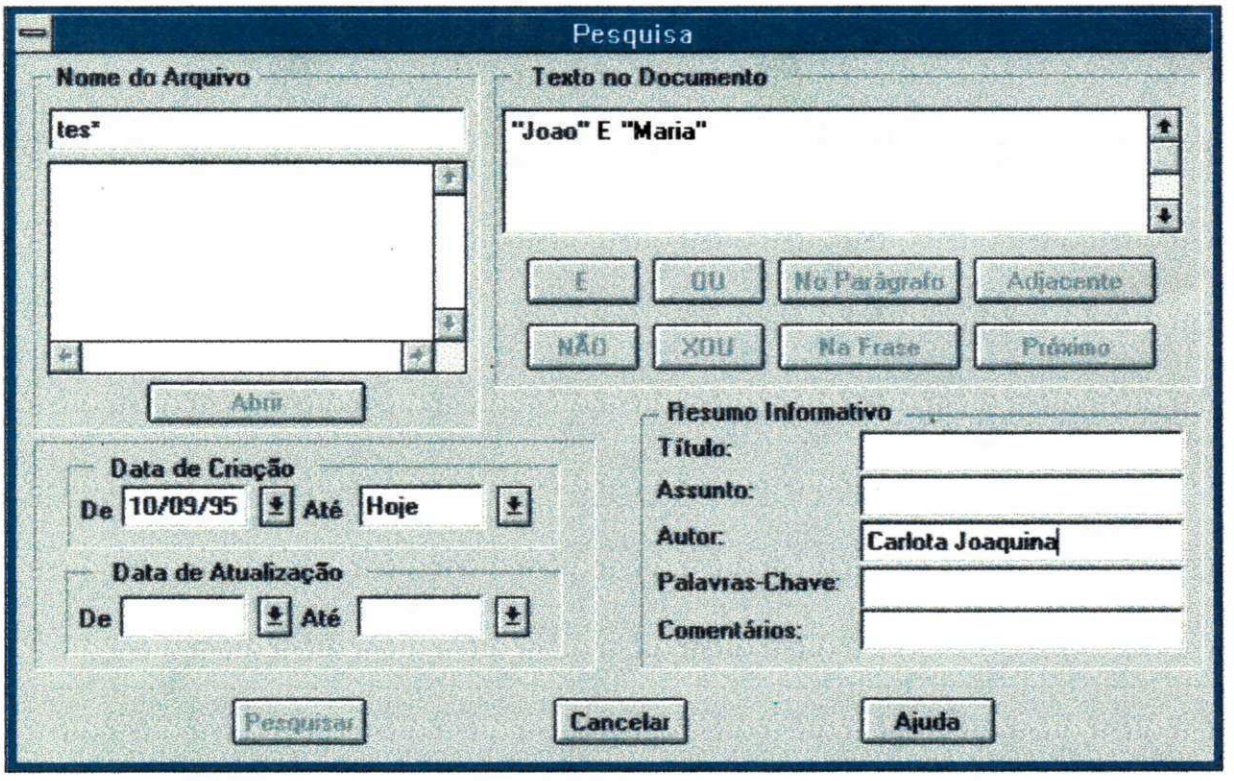
Mais uma vez, gratos pela participação ....

A.2 - Protótipo das Telas do Projeto *Add-on DocM* e Telas do Hospedeiro Word

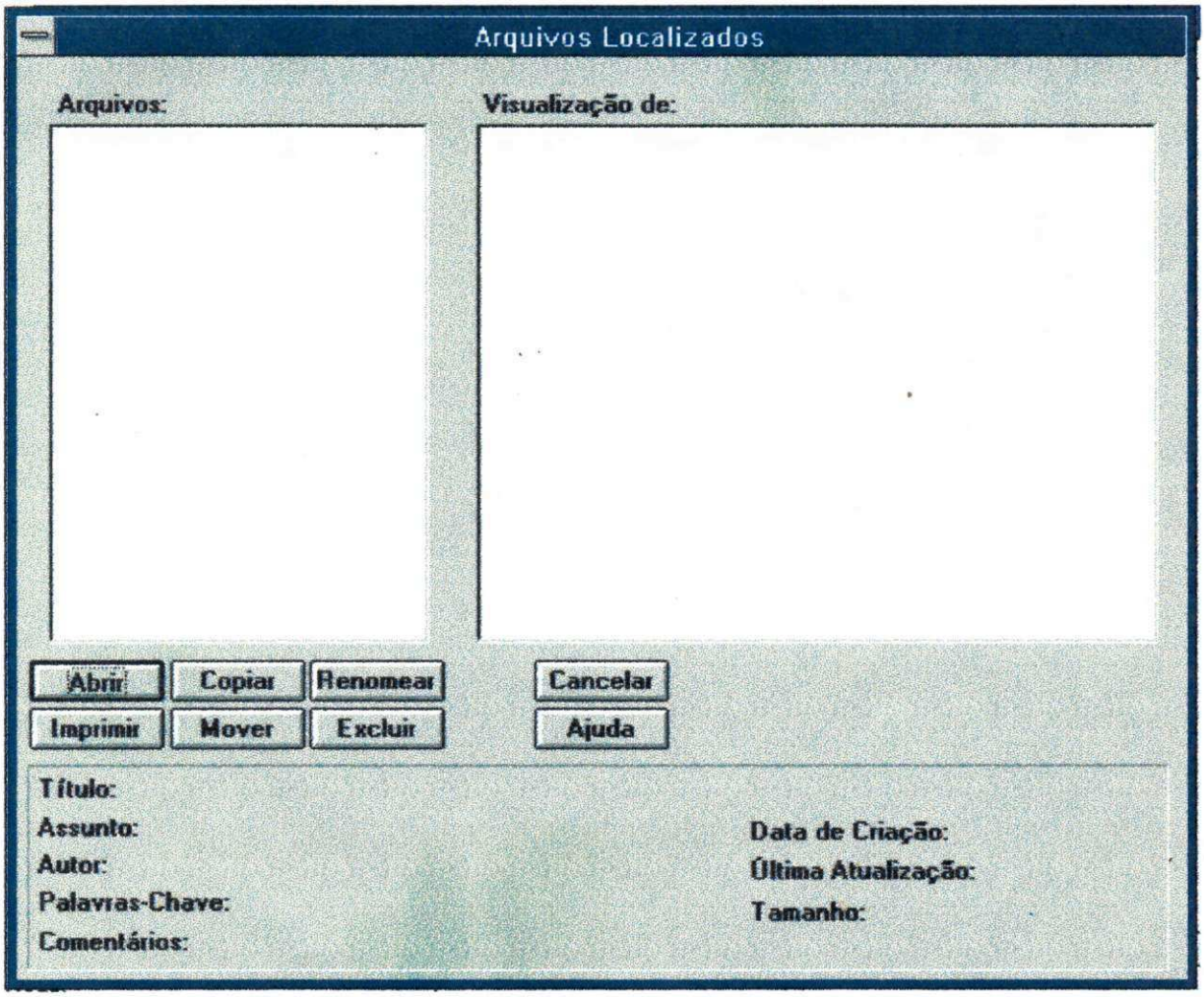
A.2.1 - Interface para Indexação dos Arquivos de Documentos



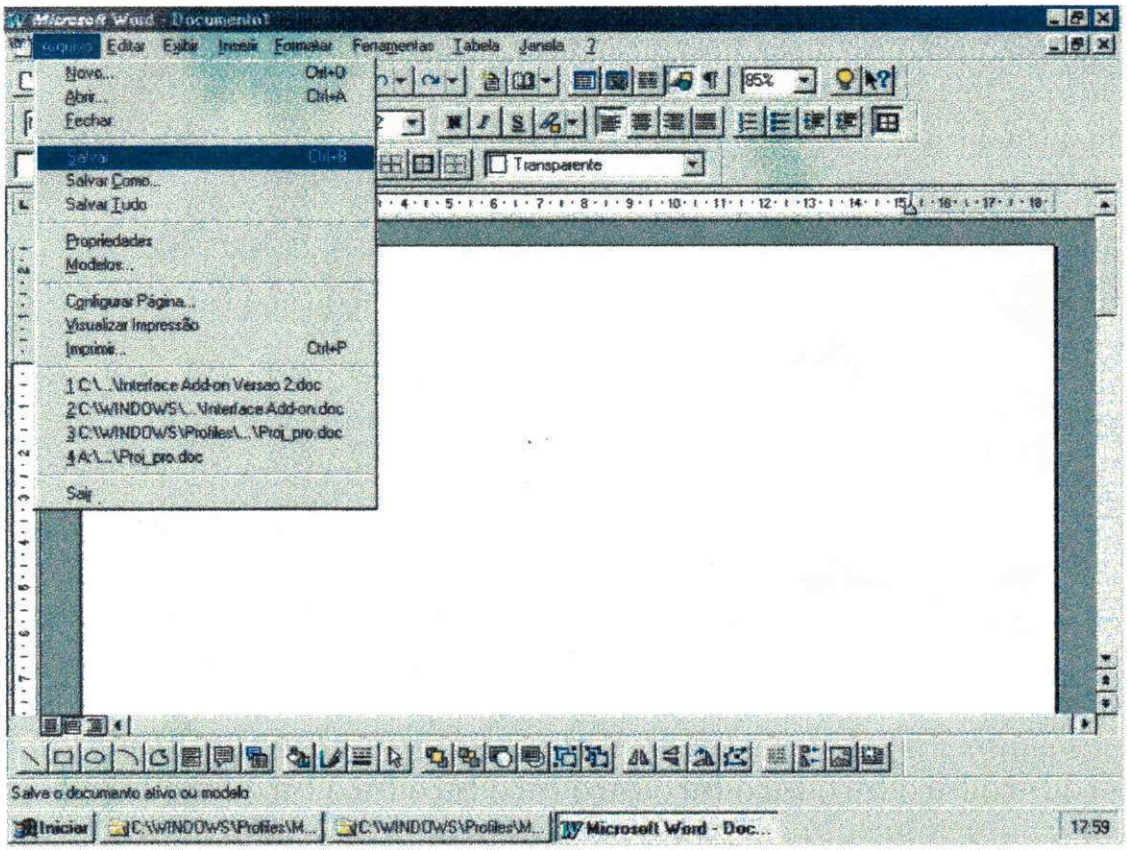
A.2.2 - Interface para Pesquisa de Documentos Previamente Indexados



A.2.3 - Interface para Visualização do Resultado de Pesquisa. Possibilita Operações Sobre os Arquivos que Compõem a Lista Final.



A.2.4 - Tela Principal da Interface do Software Hospedeiro (Word for Windows).



A.2.5 - Tela do Software Hospedeiro (Word for Windows) com uma Chamada de Pesquisa de Arquivos.