

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

(Bacharelado)

**SOFTWARE DE APOIO À AVALIAÇÃO E SELEÇÃO
DE FERRAMENTAS CASE BASEADO NA NORMA
ISO/IEC 14102**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA
COMPUTAÇÃO - BACHARELADO

JAIR WEINRICH

BLUMENAU, JUNHO/1999

1999/1-22

**SOFTWARE DE APOIO À AVALIAÇÃO E SELEÇÃO
DE FERRAMENTAS CASE BASEADO NA NORMA
ISO/IEC 14102**

JAIR WEINRICH

ESTE TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Everaldo Artur Grahl - Orientador na FURB

Prof. José Roque Voltolini da Silva - Coordenador do TCC

BANCA EXAMINADORA

Prof. Everaldo Artur Grahl

Prof. Marcel Hugo

Prof. Oscar Dalfovo

DEDICATÓRIA

Dedico esse trabalho a todos aqueles que, de alguma forma, colaboraram para o sucesso desse empreendimento de fundamental importância para o enriquecimento da minha cultura e formação profissional.

AGRADECIMENTOS

Agradeço a Deus, por manter sempre vivo em mim, um espírito de luta empenhado na busca deste sonho que agora se concretiza.

Agradeço à minha família, pela força e incentivo que deram, sempre deixando muito claro o valor de uma boa formação cultural para o ser humano.

Agradeço ao professor Everaldo Artur Grahl, pela dedicação e orientação sempre clara e precisa durante a elaboração desse trabalho.

Agradeço aos alunos da disciplina de Engenharia de Software do primeiro semestre de 1999 da FURB, que adotaram o software para avaliação de algumas ferramentas CASE.

Agradeço aos amigos, pela convivência durante essas nove fases que estivemos na FURB, pelas brincadeiras, palavras de estímulo e coragem, espírito de equipe e pela sinceridade que sempre demonstraram.

SUMÁRIO

LISTA DE FIGURAS	VIII
LISTA DE TABELAS	IX
RESUMO	XI
ABSTRACT	XII
1 INTRODUÇÃO	1
1.1 ORIGEM	1
1.2 OBJETIVO	2
1.3 ORGANIZAÇÃO.....	2
2 FERRAMENTAS CASE	4
2.1 O QUE É CASE ?.....	4
2.2 CUSTO BENEFÍCIO	7
2.3 CARACTERÍSTICAS GERAIS	9
2.4 METODOLOGIAS DE DESENVOLVIMENTO DE SISTEMAS E FERRAMENTAS CASE	10
2.5 VANTAGENS DAS FERRAMENTAS CASE.....	13
2.6 TAXONOMIA DE FERRAMENTAS CASE.....	15
2.6.1 FERRAMENTAS DE PLANEJAMENTO DE SISTEMAS COMERCIAIS.....	17
2.6.2 FERRAMENTAS DE GERENCIAMENTO DE PROJETOS.....	17
2.6.3 FERRAMENTAS DE SUPORTE.....	18
2.6.4 FERRAMENTAS DE ANÁLISE E PROJETO	19
2.6.5 FERRAMENTAS DE PROGRAMAÇÃO.....	20
2.6.6 FERRAMENTAS DE INTEGRAÇÃO E TESTE	20
2.6.7 FERRAMENTAS DE PROTOTIPAÇÃO	21
2.6.8 FERRAMENTAS DE MANUTENÇÃO.....	21

2.6.9 FERRAMENTAS DE ESTRUTURA	23
2.7 INTRODUZINDO A TECNOLOGIA CASE NA ORGANIZAÇÃO.....	23
2.8 O FUTURO DAS FERRAMENTAS CASE	27
2.9 O MERCADO DE FERRAMENTAS CASE	30
2.10 AVALIAÇÃO DE FERRAMENTAS CASE	31
3 NORMA ISO/IEC 14102 – AVALIAÇÃO E SELEÇÃO DE FERRAMENTAS CASE.....	37
3.1 ESCOPO	38
3.2 DEFINIÇÕES	38
3.3 VISÃO GERAL DA AVALIAÇÃO E SELEÇÃO DE FERRAMENTAS CASE	39
3.4 PROCESSO DE INICIAÇÃO	40
3.4.1 DEFINIÇÃO DE METAS, CRITÉRIOS DE SELEÇÃO E PLANEJAMENTO E CONTROLE DO PROJETO	41
3.5 PROCESSO DE ESTRUTURAÇÃO.....	42
3.5.1 DEFINIÇÃO DE REQUISITOS	43
3.5.2 LEVANTAMENTO DE INFORMAÇÕES SOBRE FERRAMENTAS CASE	43
3.5.3 IDENTIFICAÇÃO DAS FERRAMENTAS CASE CANDIDATAS FINAIS	44
3.6 PROCESSO DE AVALIAÇÃO.....	44
3.6.1 PREPARAR PARA A AVALIAÇÃO.....	45
3.6.2 AVALIAR FERRAMENTAS CASE.....	45
3.6.2.1 MENSURAÇÃO.....	46
3.6.2.2 PONTUAÇÃO	46
3.6.2.3 JULGAMENTO.....	46
3.6.3 RELATÓRIO DE AVALIAÇÃO	46
3.7 PROCESSO DE SELEÇÃO	47
3.7.1 PREPARAR PARA A SELEÇÃO	48
3.7.2 APLICAR O ALGORITMO DE SELEÇÃO.....	48

3.7.3 RECOMENDAR UMA DECISÃO DE SELEÇÃO.....	49
3.7.4 VALIDAR A DECISÃO DE SELEÇÃO	49
3.8 CARACTERÍSTICAS DE FERRAMENTAS CASE.....	49
3.8.1 FUNCIONALIDADE – CARACTERÍSTICAS RELACIONADAS AO PROCESSO DE CICLO DE VIDA ...	50
3.8.2 FUNCIONALIDADE – CARACTERÍSTICAS RELATADAS NO USO DA FERRAMENTA CASE.....	53
3.8.3 CARACTERÍSTICAS GERAIS DE QUALIDADE	54
4 DESCRIÇÃO DO SOFTWARE	56
4.1 INTRODUÇÃO	56
4.2 LISTA DE EVENTOS	57
4.3 DIAGRAMA DE CONTEXTO	57
4.4 DIAGRAMA DE FLUXO DE DADOS	58
4.5 MODELO ENTIDADE-RELACIONAMENTO	60
4.6 DICIONÁRIO DE DADOS.....	61
4.7 PRINCIPAIS TELAS, RELATÓRIOS E GRÁFICO DO SOFTWARE	66
4.8 APLICAÇÃO DO SOFTWARE	71
5 CONCLUSÕES	72
5.1 CONSIDERAÇÕES FINAIS	72
5.2 SUGESTÕES	74
REFERÊNCIAS BIBLIOGRÁFICAS.....	75

LISTA DE FIGURAS

1	VISÃO GERAL DA AVALIAÇÃO E SELEÇÃO DE FERRAMENTAS CASE.....	40
2	VISÃO GERAL DO PROCESSO DE INICIAÇÃO	41
3	VISÃO GERAL DO PROCESSO DE ESTRUTURAÇÃO	42
4	VISÃO GERAL DO PROCESSO DE AVALIAÇÃO	45
5	VISÃO GERAL DO PROCESSO DE SELEÇÃO.....	48
6	DIAGRAMA DE CONTEXTO.....	57
7	DIAGRAMA DE FLUXO DE DADOS - NÍVEL 1	58
8	DIAGRAMA DE FLUXO DE DADOS - NÍVEL 2	59
9	MODELO ENTIDADE-RELACIONAMENTO.....	60
10	TELA INICIAL DO SOFTWARE	67
11	MANUTENÇÃO DE FERRAMENTAS CASE	67
12	MANUTENÇÃO DAS SUBCARACTERÍSTICAS DE FERRAMENTAS CASE	68
13	MANUTENÇÃO DAS NOTAS ATRIBUÍDAS ÀS SUBCARACTERÍSTICAS DE FERRAMENTAS CASE	68
14	QUESTIONÁRIO DE AVALIAÇÃO DAS FERRAMENTAS	69
15	RELAÇÃO DAS SUBCARACTERÍSTICAS COM AS NOTAS ATRIBUÍDAS	69
16	RELAÇÃO DE FERRAMENTAS CASE COM PONTUAÇÃO OBTIDA	70
17	NÍVEL DE ATENDIMENTO DOS REQUISITOS DA ORGANIZAÇÃO	71

LISTA DE TABELAS

1	CUSTOS DE DESENVOLVIMENTO DE SISTEMAS USANDO CASE.....	7
2	ECONOMIA DE PESSOAL DE DESENVOLVIMENTO DE SISTEMAS USANDO CASE	8
3	ECONOMIA DE TEMPO DE DESENVOLVIMENTO DE SISTEMAS USANDO CASE	8
4	EVOLUÇÃO DAS METODOLOGIAS DE DESENVOLVIMENTO DE SISTEMAS	11
5	COMPARAÇÃO ENTRE DESENVOLVIMENTO DE SOFTWARE DO MODO TRADICIONAL E ATRAVÉS DO USO DE METODOLOGIAS.....	12
6	CARACTERÍSTICAS RELACIONADAS AO PROCESSO DE CICLO DE VIDA	50
7	CARACTERÍSTICAS RELATADAS NO USO DA FERRAMENTA CASE.....	53
8	CARACTERÍSTICAS GERAIS DE QUALIDADE.....	54
9	TABELA DE FERRAMENTAS.....	61
10	TABELA DE FORNECEDORES	62
11	TABELA DE AVALIADORES	63
12	TABELA DE CARACTERÍSTICAS	63
13	TABELA DE SUBCARACTERÍSTICAS	63
14	TABELA DE AVALIAÇÕES	64
15	TABELA DE CONFIGURAÇÕES DE HARDWARE	64
16	TABELA DE REQUISITOS	64
17	TABELA DE METAS.....	65
18	TABELA DE SUBCARACTERÍSTICAS DA NORMA ISO/IEC 14102	65
19	TABELA DE FASES DO CICLO DE VIDA DA ENGENHARIA DE SOFTWARE.....	65
20	TABELA DE LINGUAGENS DE PROGRAMAÇÃO	65
21	TABELA DE METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE	66
22	TABELA DE SISTEMAS OPERACIONAIS	66

23	TABELA DE BANCOS DE DADOS	66
24	TABELA DE SOFTWARES.....	66

RESUMO

A utilização de metodologias e ferramentas CASE para o desenvolvimento e manutenção de software, proporcionam incremento de produtividade e melhoria da qualidade nos softwares produzidos. Parte fundamental desse princípio é a escolha da ferramenta ou conjunto delas que melhor atende às necessidades da equipe de desenvolvimento. Esse trabalho apresenta um software baseado na norma ISO/IEC 14102 – Avaliação e seleção de ferramentas CASE. Esta norma define tarefas e procedimentos através de seus processos, que conduzem à seleção da ferramenta que atende aos requisitos da organização referentes ao desenvolvimento de software. O software desenvolvido tem por objetivo apoiar a avaliação e seleção de ferramentas CASE, através da utilização de questionários.

ABSTRACT

The use of methodologies and CASE tools for the development and software maintenance, provide productivity increment and improvement of the quality in the produced softwares. Fundamental part of that beginning is the choice of the tool or set of them that best assists to the needs of the development team. That work presents a software based on the norm ISO/IEC 14102 - Evaluation and selection of CASE tools. This norm defines tasks and procedures through its processes, that lead to the selection of the tool that assists to the requirements of the organization referring to the software development. The developed software has for objective to support the evaluation and selection of CASE tools, through the use of questionnaires.

1 INTRODUÇÃO

1.1 ORIGEM

Nesses tempos de globalização, uma das áreas que mais têm evoluído na economia mundial por seu valor estratégico é o desenvolvimento e a comercialização de software. Para ser competitivo neste mercado especializado, deve-se dispor de um bom conjunto de ferramentas e recursos computacionais. Nesse contexto, destacam-se as ferramentas CASE (*Computer-Aided Software Engineering* - Engenharia de Software Auxiliada por Computador), que na área da Engenharia de Software, representam a maior parte das tecnologias de suporte usadas para o desenvolvimento e manutenção de sistemas [ISO98].

Nos últimos anos, verifica-se que as ferramentas disponíveis aos engenheiros de software ou desenvolvedores de sistemas evoluíram significativamente, distanciando-se muito das originalmente concebidas que, quando muito, disponibilizavam compiladores e editores de textos. Dentre essas, deve-se considerar que as ferramentas CASE são as que mais evoluíram, apresentando uma série de recursos antes inexistentes [PRE95].

Considerando essas ferramentas como fator decisivo para a obtenção de sucesso na atividade de desenvolvimento e manutenção de software, há de se convir que uma seleção acertada por parte das organizações que irão utilizá-las, se faz necessária. Várias ferramentas estão presentes no mercado e pode-se constatar que elas diferem entre si, mesmo que por alguma sutil diferença.

No intuito de facilitar o momento da escolha da ferramenta que melhor atende às necessidades e os requisitos das organizações que constroem software, foi desenvolvido esse trabalho baseado na norma ISO/IEC 14102 – Avaliação e seleção de ferramentas CASE. Este trabalho é importante porque facilita a execução de algumas tarefas propostas pela norma, como por exemplo, a montagem do questionário de avaliação, o registro das notas atribuídas a cada subcaracterística da ferramenta, a execução do algoritmo de acumulação da pontuação obtida pela avaliação das ferramentas, emissão de um relatório com os dados básicos e a

totalização obtida por cada ferramenta durante o processo de avaliação. Isto facilita o processo de seleção, que no trabalho proposto, se dará pela maior pontuação alcançada.

É importante salientar que antes de adquirir uma ferramenta CASE, uma organização deverá estar familiarizada com alguma metodologia de desenvolvimento de software para só então ir ao mercado e selecionar uma ferramenta. Desse modo, ela poderá usufruir de todos os benefícios que a adoção da tecnologia CASE trará à organização.

Algumas organizações acreditam que a adoção da tecnologia CASE irá resolver todos os seus problemas de produtividade e qualidade de software, porém isso nem sempre ocorre. Como anteriormente mencionado, sem a utilização consciente de uma metodologia de desenvolvimento de software, adotar ferramentas CASE pode ser considerado como apenas automatizar a desordem, o que não resolve em nada os problemas da organização.

1.2 OBJETIVO

O objetivo deste trabalho é especificar e implementar um software de apoio à avaliação e seleção de ferramentas CASE, seguindo procedimentos, recomendações e critérios baseados na norma ISO/IEC 14102 – Avaliação e seleção de ferramentas CASE.

1.3 ORGANIZAÇÃO

A seguir é apresentada uma síntese dos capítulos constantes desse trabalho.

O capítulo de introdução apresenta uma visão geral do presente trabalho, o contexto em que está inserido, sua importância e objetivo.

O segundo capítulo apresenta uma fundamentação ao tema ferramentas CASE, demonstrando a importância do seu uso, relação custo/benefício, a relação existente entre metodologias e CASE, vantagens, tipos de ferramentas existentes, como introduzir a tecnologia CASE numa organização, o que esperar do futuro da CASE, um estudo sobre o mercado e encerra o capítulo, apresentando formas de melhor avaliar e selecionar ferramentas CASE.

O terceiro capítulo apresenta a norma ISO/IEC 14102 – Avaliação e seleção de ferramentas CASE com sua estrutura composta por quatro processos: iniciação, estruturação, avaliação e seleção.

O quarto capítulo apresenta o software desenvolvido para esse trabalho. Descreve o software, suas características, sua especificação, principais telas e relatórios. Apresenta também sua aplicação na avaliação de algumas ferramentas CASE.

O quinto capítulo apresenta as conclusões desse trabalho e as sugestões para que o mesmo possa ter continuidade e seja melhorado.

2 FERRAMENTAS CASE

2.1 O QUE É CASE ?

A denominação CASE (*Computer-Aided Software Engineering*, ou Engenharia de Software Auxiliada por Computador) sugere a abrangência que uma ferramenta com essa designação deve possuir. Segundo [FUG93], [JOA93] e [RUT96], a denominação CASE foi criada no início da década de 1980, quando evidenciou-se a idéia de que ferramentas gráficas, como os Diagramas de Fluxos de Dados (DFD), Modelos de Entidades e Relacionamentos (MER) e Gráficos Estruturados (GE), poderiam ser úteis para as fases de análise e projeto de sistemas.

Conforme [FEL88], é notável o desenvolvimento do mercado CASE, sendo visivelmente aparente o esforço dos fabricantes dessas ferramentas em aperfeiçoar e implementar cada vez mais recursos e facilidades para atender às necessidades de planejamento, análise, projeto, construção e manutenção de sistemas, tornando possível agilizar o desenvolvimento de projetos de software, ocasionando maior rapidez e melhores resultados.

Para [FIS90], as ferramentas CASE auxiliam os engenheiros e desenvolvedores de software especificar e projetar software. Muitas ferramentas CASE vão ainda mais longe, pois geram automaticamente o software, com base nessa especificação e projeto. Seguindo essa mesma linha de pensamento, [MAR91] afirma que as ferramentas CASE são essenciais para a obtenção de apoios automatizados para o projeto de sistema, verificação computadorizada do projeto e geração automática de código a partir do projeto.

No trabalho de [JOA93], ferramentas CASE são definidas como um conjunto integrado de ferramentas que oferecem impulso, atuando em todas as fases do ciclo de desenvolvimento de projeto de software, que por conseguinte, causam um impacto profundo e necessitam de novas metodologias de desenvolvimento de sistemas também causando impactos culturais não menos profundos nos profissionais e nas organizações de desenvolvimento de software.

Uma definição simples e ao mesmo tempo completa é dada por [CLU89], ao afirmar que CASE é a automação do desenvolvimento de software. Nessa concepção, CASE oferece aos desenvolvedores de software uma abordagem diferente para o ciclo de vida de software, sendo essa, baseada na automação. A idéia básica é a existência de um conjunto integrado de ferramentas que possibilitam a economia de trabalho, unindo e automatizando todas as fases do ciclo de vida de software.

Não existe um consenso ou uma definição formal sobre o que representa a Engenharia de Software computadorizada, porém a maioria do pessoal ligado à computação concorda que a tecnologia CASE trabalha com todos os processos do ciclo de vida da Engenharia de Software, tais como, especificação de projeto, análise de requisitos e implementação do software, inclusive com a geração automática do programa executável.

Conforme [FIS90], as ferramentas CASE, quando não eliminam, reduzem drasticamente problemas de projeto e implementação, próprios de grande projetos de software, através da geração automática de código a partir da especificação do projeto e da análise dos requisitos do usuário. Com isso, o desenvolvedor de software, pode concentrar-se mais na arquitetura do sistema, pois a Engenharia de Software computadorizada concentra-se mais nessa fase, onde um trabalho bem realizado garante uma geração de código eficaz, enxuta, de qualidade, documentada e de acordo com o que foi inicialmente projetado.

Nos tempos atuais, a tecnologia CASE faz parte do conjunto de ferramentas do engenheiro de software, proporcionando-lhe a capacidade de automatizar as diversas fases do ciclo de desenvolvimento de software. Conjunto esse, de ferramentas semi-automatizadas e automatizadas, que implementa aos profissionais e às organizações, uma cultura de engenharia que é novidade para a maioria do pessoal envolvido pela tecnologia CASE.

Um fator primordial que justifica ainda mais o uso de tecnologia CASE é o fato de que os projetos de software vêm crescendo em complexidade, tornando-se propensos a comportamentos imprevisíveis, podendo chegar até ao fracasso total. Esse problema surge devido a enorme quantidade de software necessária para atender aos sistemas de aplicativos desenvolvidos em nossa época atual. Quanto mais crescem os projetos de software, maior a probabilidade de ocorrerem erros, pois deve-se considerar que muitos projetos possuem até alguns milhares de linhas de código, podendo, em alguns casos chegar a milhões de linhas.

A tecnologia de ferramentas CASE também se aplica muito bem nas organizações quando há falta de pessoal capacitado para o desenvolvimento de software, fator esse até bastante comum. As filas de espera de desenvolvimento de software (*backlogs*), muitas vezes ocorrem devido a demora na fase de implementação do aplicativo, bem como a dificuldade, muitas vezes de contratar desenvolvedores qualificados. Com a adoção de CASE, a fila de espera tende a reduzir pois torna mais produtiva a hora de trabalho de cada profissional.

Segundo [FIS90], estudos realizados na década de 1980, revelaram que 64% dos problemas ocorridos nos softwares, tinham sua origem nas fases de especificação de projetos e análise de requisitos, sendo que apenas 30% dessa enorme quantidade de erros era detectada antes que o software passasse para a fase de testes de aprovação. Por outro lado, durante a fase de implementação do projeto, 36% dos erros eram decorrentes de problemas de programação, sendo que 75% desses mesmos erros de codificação eram encontrados antes do início dos testes de aprovação. Levando-se em consideração esse estudo, constata-se a necessidade de adoção da tecnologia de ferramentas CASE, pois é necessário projetar corretamente já na primeira vez.

Um fator que pode oferecer resistência à adoção de ferramentas CASE, é o fato de que muitos desenvolvedores de software consideram-se verdadeiros artistas, a partir de um raciocínio solidamente fundamentado nos primórdios da era da computação sendo que, devido a esse fator, acabam visualizando seu trabalho como uma forma de arte. Numa primeira análise, as ferramentas CASE tendem a diminuir a aura artística desses profissionais ao oferecer ferramentas que sugerem seguir um regime severo de desenvolvimento de software. Muitos desses profissionais acreditam que a criatividade em Engenharia de Software é a essência da arquitetura de um programa bem elaborado que atenda as necessidades do usuário. Deve-se deixar sempre bem claro que a verdadeira arte reside na implementação da interface com o usuário e na criação e organização das estruturas internas, bem como suas interconexões. Segundo [FIS90], as ferramentas CASE não eliminarão a criatividade dos desenvolvedores de software, pelo contrário, oferecerão ferramentas para demonstrá-la melhor, pois possibilitam a construção de projetos de software mais elaborados em menos tempo, devido ao fato de que essas ferramentas na sua grande maioria, apresentam uma forma de desenvolver software de modo gráfico o que estimula ainda mais o lado criativo dos profissionais das equipes de desenvolvimento.

2.2 CUSTO BENEFÍCIO

Um outro fator fundamental levado em consideração quando da adoção de qualquer tipo de tecnologia, é sua relação custo/benefício. Com a tecnologia de ferramentas CASE não poderia ser diferente. Tem-se observado e comprovado que diminuem drasticamente o custo de desenvolvimento de sistemas com o uso de ferramentas CASE, conforme pode-se verificar na tabela 1.

Tabela 1 : Custos de desenvolvimento de sistemas usando CASE

Tamanho da aplicação (em pontos por função)	Custo atual em US\$	Custo com a adoção de CASE em US\$	Incremento de produtividade
1	375	15	25:1
10	5.000	250	20:1
100	125.000	8.000	15:1
1.000	3.000.000	250.000	12:1
10.000	80.000.000	10.000.000	8:1
100.000	1.250.000.000	250.000.000	5:1

Fonte: [JOA93] - YOURDON; Software Methodologies, An Executive Briefing

Analisando a tabela 1, pode-se verificar que os valores foram colocados de maneira especulativa, porém servem para ressaltar de maneira realista os efeitos de uma verdadeira automação nos processos de desenvolvimento de software, utilizando CASE.

Outro fator importante, quando da análise do uso ou não de ferramentas CASE, é a aplicação da quantidade de pessoal necessário no desenvolvimento de software. Embora para o desenvolvimento de pequenas aplicações ainda se necessite de um mínimo de um profissional, essa relação muda significativamente quando ocorre a necessidade de se desenvolver grandes aplicações. Para que o custo com pessoal não extrapole valores desejáveis para o custo global do projeto, torna-se fundamental o uso da tecnologia CASE a fim de minimizar esse custo para poder ter preços competitivos no mercado, mas não diminuindo a qualidade do software, o que garante a satisfação do usuário final da aplicação. Uma comparação do uso de profissionais de desenvolvimento de aplicações utilizando ferramentas CASE é apresentada na tabela 2.

Tabela 2 : Economia de pessoal de desenvolvimento de sistemas usando CASE

Tamanho da aplicação (em pontos por função)	Pessoal atual	Pessoal com CASE	Incremento de produtividade
1	1	1	1:1
10	1	1	1:1
100	2	1	2:1
1.000	16	8	2:1
10.000	260	165	1,6:1
100.000	2.000	1.400	1,4:1

Fonte: [JOA93] - YOURDON; Software Methodologies, An Executive Briefing

Nota-se analisando a tabela 2, que para pequenas aplicações, de tamanho entre 1 e 10 pontos por função (métrica utilizada para analisar o tamanho e a complexidade do software desenvolvido), mesmo com a utilização de ferramentas CASE, sempre será necessário, pelo menos um profissional de desenvolvimento, pois a ferramenta, por mais que incremente a produtividade, ela nunca poderá trabalhar sozinha.

Nos tempos atuais, em que a escassez de tempo é cada vez maior, é imprescindível desenvolver software em menos tempo, com o propósito de diminuir custos globais de desenvolvimento e atender aos usuários o mais breve possível, fornecendo da mesma maneira, software de qualidade a um baixo custo. Essa redução de tempo de desenvolvimento e de custo de projeto, pode ser obtida com o uso da tecnologia CASE, conforme [JOA93] apresenta na tabela 3.

Tabela 3 : Economia de tempo de desenvolvimento de sistemas usando CASE

Tamanho da aplicação (em pontos por função)	Pessoal atual	Pessoal com CASE	Incremento de produtividade
1	1,25 dia	30 minutos	25:1
10	1 mês	1 dia	20:1
100	1 ano	1 mês	12:1
1.000	3 anos	6 meses	6:1
10.000	5 anos	1 ano	5:1
100.000	10 anos	3 anos	3,3:1

Fonte: [JOA93] - YOURDON; Software Methodologies, An Executive Briefing

Através da análise da tabela 3, pode-se constatar uma relação inversa entre tamanho da aplicação e incremento de produtividade. Quanto menor for a aplicação, com o uso de tecnologia CASE, maior será o incremento da produtividade no desenvolvimento de software, sendo que esse incremento diminui à medida que a aplicação cresce em complexidade e tamanho.

2.3 CARACTERÍSTICAS GERAIS

As ferramentas CASE necessitam atender alguns requisitos fundamentais para fazerem parte das ferramentas dos desenvolvedores de software. Isso é importante para que se obtenham maiores benefícios quando da adequação e implementação das ferramentas às metodologias de desenvolvimento das organizações.

Segundo [FIS90], as ferramentas CASE devem realizar as seguintes tarefas:

- a) fracionamento da complexidade: uma das metas principais da tecnologia CASE é decompor os requisitos e os projetos em componentes manejáveis. Sua função é simplificar, explicar e reduzir;
- b) adequação a um público diversificado: para as fases de requisitos e de projeto do ciclo de desenvolvimento de software, as ferramentas CASE servem a diversas pessoas. Por um lado, sua saída deve ser inteligível para os usuários finais e as organizações contratantes, que pagam pelo desenvolvimento do software. Por outro lado, devem oferecer uma ajuda aos desenvolvedores; caso contrário, é perda de tempo utilizá-las;
- c) mais baratas que a construção em si: a utilização de uma ferramenta CASE deve custar menos e ser mais eficaz a longo prazo do que o desenvolvimento de um sistema pelos métodos tradicionais. As ferramentas CASE devem reduzir substancialmente o empenho despendido em implementação e manutenção, oferecendo especificações e projetos de qualidade superior;
- d) quantitativas e verificáveis: as especificações e projetos gerados pelas ferramentas CASE devem articular as características e os componentes do software a ser construído. Cada exigência da implementação tem que ser verificável e poder ser encontrada no documento dos requisitos. Os critérios de

- desempenho, as limitações e as condições de erro devem estar especificadas no projeto;
- e) de fácil manutenção: as especificações e projetos produzidos por uma ferramenta CASE devem ser adaptáveis às modificações nas metas dos requisitos e dos projetos. Quando um documento do projeto perde a sincronização com o código subordinado, torna-se inútil e pode até causar perda de tempo aos desenvolvedores em futuros aperfeiçoamentos do software;
 - f) orientação gráfica: as boas ferramentas CASE apresentam informações visuais de especificações e projeto. São para a Engenharia de Software o que os programas CAD (*Computer-Aided Design* - Projeto Auxiliado por Computador) são para a engenharia civil. Tanto para os usuários finais como para os desenvolvedores, é muito mais fácil compreender uma ilustração gráfica do que ler inúmeras páginas de texto descritivo.

2.4 METODOLOGIAS DE DESENVOLVIMENTO DE SISTEMAS E FERRAMENTAS CASE

Ferramentas CASE implementam e dão suporte a pelo menos uma metodologia de desenvolvimento de sistemas. A maioria das ferramentas atuais implementam a metodologia de Análise Estruturada e a metodologia de Análise Orientada a Objetos. Devido a essa realidade, nota-se que tentar adotar a tecnologia de ferramentas CASE sem que a organização possua uma sólida fundamentação em metodologias de desenvolvimento de software, é bastante improvável que se obtenha sucesso e os benefícios que a CASE pode oferecer.

A fim de melhor entender a união entre as metodologias desenvolvidas durante a história da computação e conseqüentemente sua adoção por parte das ferramentas CASE, faz-se necessário conhecer a própria evolução dessas metodologias. Uma evolução cronológica aproximada das metodologias de desenvolvimento de sistemas é apresentada por [JOA93] na tabela 4. Através da análise da tabela, verifica-se que a bastante tempo existe a preocupação em se desenvolver e utilizar metodologias, visto que por volta do ano de 1965, desenvolveu-se a metodologia de Análise Estruturada. Durante um longo período, essa metodologia dominou o cenário de desenvolvimento de software, até que por volta do ano de 1990, evidenciou-se com maior ênfase, a utilização da metodologia de Análise Orientada a Objetos.

Tabela 4 : Evolução das metodologias de desenvolvimento de sistemas

Ano (época)	Metodologias (técnicas) de desenvolvimento de sistemas
1965	Metodologia estruturada
1970	Técnicas de modelagem de dados
1975	Projeto de bancos de dados Bancos de dados e linguagens de quarta geração
1980	Especificação do projeto Ferramentas de software Modelagem de dados Linguagens de quarta geração Prototipação
1985	Interface com o usuário Ferramentas de prototipação Automação das metodologias (ferramentas CASE)
1990	Ferramentas de geração de código Metodologias orientadas a objeto

Fonte: [JOA93] - Metodologias de desenvolvimento de sistemas

Fator importante é a escolha de uma metodologia de desenvolvimento, pois ela poderá se revelar inapropriada para automação através do uso de ferramentas CASE. Caso a metodologia escolhida seja manual, o descontentamento com o uso da metodologia poderá migrar também para o uso da CASE e a organização não aproveitará todos os benefícios que resultam da escolha de uma metodologia adequada juntamente com ferramentas CASE que possibilitem implementar tal metodologia.

Devido a importância que deve ser dada a escolha de uma metodologia de desenvolvimento de sistemas adequada, faz-se necessário conhecer um pouco mais acerca do seu uso e as conseqüências quando da sua não utilização. Um exemplo de como metodologias podem ser úteis para resolver problemas comuns frente ao desenvolvimento tradicional de sistemas, é apresentado por [MAR83] na tabela 5. A partir da utilização de metodologias, se obtém algumas vantagens como por exemplo, diminuição do tempo total do projeto, proporcionando uma redução importante nos custos de desenvolvimento, bem como a maioria dos erros são encontrados antes da liberação do software para o usuário.

Tabela 5 : Comparação entre desenvolvimento de software do modo tradicional e através do uso de metodologias

Problemas com o desenvolvimento tradicional de sistemas	Os efeitos da aplicação de metodologias no desenvolvimento de sistemas
a) o desenvolvimento leva muito tempo	a) o desenvolvimento é muito mais rápido
b) os custos de desenvolvimento são muito altos	b) uma redução importante nos custos de desenvolvimento
c) a programação é manual	c) a programação pode ser automatizada através do uso de ferramentas CASE
d) a maioria dos erros são encontrados após a implementação	d) a maioria dos erros são encontrados antes da implementação
e) a maioria dos erros são encontrados manualmente	e) a maioria dos erros são encontrados por análise automática e estática
f) alguns erros nunca são encontrados	f) a maioria dos erros são encontrados
g) desencontro entre requisitos e especificações	g) cada nível de expansão precisa do nível anterior
h) desencontro entre especificações e projeto	h) desencontro entre o projeto e a implementação
i) especificações incompletas e inconsistentes	i) especificações internamente completas e consistentes são reforçadas
j) muitos erros de interface e desencontros entre sistemas	j) interfaces rigorosas e comprovadamente corretas entre os subsistemas
k) nenhuma garantia de integridade da função após a implementação	k) garantia da integridade da função após a implementação
l) grandes equipes de desenvolvimento com problemas sérios de comunicação	l) equipes menores ou de uma pessoa com menos problemas de comunicação
m) grande papelada para o controle administrativo	m) eliminação da maioria da papelada
n) muito desenvolvimento de código redundante	n) identificação de módulos comuns sendo sua utilização feita mais facilmente
o) pessoal de desenvolvimento separado reinventando a roda	o) mecanismo de biblioteca de interfaces rigorosas encoraja construção de estruturas reutilizáveis
p) dificuldade de manutenção	p) facilidade de manutenção
q) as modificações disparam reações em cadeia ocasionando novos erros	q) os efeitos de todas as modificações são feitos de forma clara e objetiva
r) manutenção sucessiva deteriora a qualidade do código	r) código de alta qualidade e regenerado após cada modificação
s) problemas na portabilidade	s) um projeto pode ser regenerado para diferentes ambientes

Fonte: [MAR83] - Metodologias de desenvolvimento de sistemas

Um fator importante no desenvolvimento de sistemas é a adoção de uma metodologia de desenvolvimento de sistemas, que tem por objetivo identificar e estruturar todos os processos e tarefas necessários à implementação de um software de qualidade. Sem a adoção dessa metodologia, não é aconselhável a adoção de tecnologia de ferramentas CASE, pois se estaria apenas automatizando a desordem, não provendo à equipe desenvolvedora os reais benefícios que são obtidos com a união de uma metodologia de desenvolvimento de sistemas com a tecnologia CASE.

2.5 VANTAGENS DAS FERRAMENTAS CASE

Para a grande maioria das organizações de desenvolvimento de software, as vantagens qualitativas obtidas através do uso de ferramentas CASE têm um peso maior que as vantagens quantitativas. Conforme [FIS90], o tempo gasto no desenvolvimento será quase sempre menor com o auxílio das ferramentas CASE, mas talvez seu maior benefício tenha a forma de garantia, a consciência tranqüila de que a tarefa está sendo realizada devidamente como foi programada e seguindo as especificações do usuário.

Entre as diversas vantagens que se obtêm com o uso de ferramentas CASE, destacam-se conforme [CHI93], [COO94] e [FIS90]:

- a) especificações completas dos requisitos: especificar completamente os requisitos do sistema é o objetivo das ferramentas CASE de análise e de especificação dos requisitos. A maioria das metodologias de especificação impõe o envolvimento do usuário final com o objetivo de desenvolver um modelo. Embora ainda exista o risco de se criar uma grande confusão, apesar do maior empenho de todos os envolvidos, essa probabilidade é bem menor, se houver uma especificação completa, detalhada e minuciosa dos resultados;
- b) especificações minuciosas do projeto: documentar as especificações do projeto de forma completa e precisa é necessário para o entendimento, desenvolvimento e manutenção do software. Além disso, a documentação deve recomendar a separação de temas em módulos distintos, cada um com entradas e saídas bem definidas, onde cada módulo é tratado como uma caixa preta, cuja estrutura interna de implementação não é conhecida do público. Apenas os parâmetros de entrada e saída são visíveis aos demais módulos;

- c) especificações atuais do projeto: as ferramentas CASE para projetos ajudam a manter uma sincronia com a implementação do código. Muitas delas vinculam de fato o código à especificação, de modo que, se a especificação se modificar, o código a ela subordinado também se modifique. Para as ferramentas CASE atuais, que na sua maioria geram código automaticamente, não há necessidade de se manter a sincronização, pois o desenvolvedor não toca no código, apenas edita o projeto do sistema e a implementação é gerada ou regerada automaticamente. As ferramentas CASE oferecem dispositivos para manterem as especificações do projeto atualizadas em face de manutenção contínua do software, redefinição de projeto, aprimoramentos e evoluções de implementação;
- d) redução do tempo de desenvolvimento: a especificação completa da arquitetura do software visa eliminar a perda de tempo em implementações equivocadas, sendo que esse ganho de produtividade se traduz em tempo de implementação reduzido. Por causa da orientação gráfica altamente interativa da maioria das ferramentas CASE, utilizar uma delas leva muitos técnicos de software a acreditar que estão escrevendo o código, quando na realidade estão fazendo os projetos arquitetônicos do software. O tempo gasto na elaboração do projeto é mais do que compensado nas fases de implementação, testes e lançamento;
- e) código altamente flexível e de fácil manutenção: nenhum projeto de software bem sucedido chega ao fim. Os usuários finais vão exigir ou sugerir aperfeiçoamentos funcionais ou identificarão erros na operação do software, tornando necessária alguma forma de desenvolvimento contínuo ou de serviço de manutenção. Embora os pequenos aperfeiçoamentos ou reparos de erros possam não garantir uma atualização da especificação do projeto, a combinação de diversas modificações o fará. Torna-se difícil até para as ferramentas CASE, dar assistência na manutenção das especificações do projeto, se o software desenvolvido a partir dessas especificações estiver escrito à mão e não a partir da geração automática da implementação pela própria ferramenta. Felizmente, ferramentas que geram código automaticamente a partir da especificação do projeto de software, não se atrapalham com esse problema.

Em resumo, as ferramentas CASE de alta qualidade:

- a) são mais atraentes de se usar do que implementar software à mão;

- b) ajudam mais a projetar do que a documentar;
- c) mantêm a sincronização projeto-implementação;
- d) reduzem o risco de falhas e surpresas;
- e) reduzem o tempo total do desenvolvimento de software.

Quando se descreve as vantagens do uso da tecnologia CASE, corre-se sempre o risco de que alguma facilidade, vantagem ou característica importante dessas ferramentas fique à margem das observações e descrições. Uma importante justificativa para o uso de ferramentas CASE, ressaltando a vantagem de seu uso, é apresentada por [FOU94] que recomenda firmemente o uso de diversas ferramentas automatizadas ao longo das fases do ciclo de vida de desenvolvimento e manutenção.

2.6 TAXONOMIA DE FERRAMENTAS CASE

Segundo [PRE95], as ferramentas CASE podem ser classificadas por função, por seus papéis como instrumentos para os gerentes e para o pessoal técnico, pelo uso que elas têm nas várias etapas do processo de Engenharia de Software, pela arquitetura de ambiente – hardware e software que as suporta, ou até mesmo pela origem ou custo delas.

Uma taxonomia simplificada de ferramentas CASE destina-se a classificar as ferramentas de acordo com suas funções no ciclo de desenvolvimento do software onde as ferramentas acrescentam impulso, como por exemplo, nas fases de especificação de projeto, análise de requisitos ou implementação de código, ressalta [FIS90]. Porém deve-se considerar que algumas ferramentas implementam mais que uma fase ou processo do ciclo de desenvolvimento e uma tendência natural das ferramentas CASE, é cada vez mais abranger e lidar com todas as fases do ciclo de desenvolvimento de software.

Diferentes abordagens de categorização de ferramentas CASE tem surgido nos últimos anos, pois a tecnologia CASE tem sido evidenciada cada vez mais, fazendo parte da cultura de desenvolvimento de software de muitos analistas de sistemas e engenheiros de software. O mercado CASE tem se desenvolvido rapidamente nos últimos anos com alto grau de expansão. Com esse crescimento muitas vezes desordenado, a CASE constantemente está em processo de mutação e não há um padrão definido para sua categorização embora, após o

seu surgimento, vários autores tentaram categorizá-las. No entanto, pode-se classificar ferramentas CASE como citado em [JOA93]:

- a) *front end* ou *upper CASE*: são aquelas que apoiam as etapas iniciais de criação dos sistemas, são elas as fases de planejamento, análise e projeto do software;
- b) *back end* ou *lower CASE*: são aquelas que dão apoio à parte física, isto é, à codificação, testes e manutenção da aplicação;
- c) *i-CASE* ou *integrated CASE*: classifica os produtos que cobrem todo o ciclo de desenvolvimento de software, desde a fase de especificação de projeto e análise de requisitos até o controle final da qualidade do software implementado.

Sob a ótica da fase do ciclo de vida de software que a ferramenta CASE apoia, apresenta-se a categorização, conforme segue:

- a) *planning workstation*: uma estação de trabalho que tem por função automatizar o planejamento do sistema da organização, apoiando as técnicas de planejamento estratégico de informações;
- b) *analysis workstation*: tem por objetivo apoiar e auxiliar o analista de sistemas no seu trabalho de especificação formal e rigorosa de software, seguindo técnicas de análise de sistemas;
- c) *design workstation*: é voltada ao projetista de sistemas que auxilia no projeto da arquitetura de código do software, seguindo métodos de desenvolvimento amplamente divulgados, testados e recomendados, como por exemplo, a metodologia de Yourdon/DeMarco;
- d) ferramentas de programação: são as ferramentas destinadas à implementação de software. Sua funcionalidade maior é obtida em ambientes integrados com outras ferramentas, como exemplo, ferramentas de especificação de projeto e análise de requisitos.

Segue-se apresentando outra taxonomia de ferramentas CASE que tenta melhor categorizar as ferramentas presentes no mercado, servindo de base para a classificação e distinção entre as diversas ferramentas em uso atualmente pelos profissionais de desenvolvimento de software, conforme [PRE95].

2.6.1 FERRAMENTAS DE PLANEJAMENTO DE SISTEMAS COMERCIAIS

Essas ferramentas são utilizadas para modelar os requisitos de informação estratégicos da organização. Através desta categoria de ferramentas a informação é modelada, não se restringindo aos requisitos de uma aplicação específica. Sua função principal é disseminar e melhorar a compreensão do modo como as informações fluem dentro da organização, através de suas unidades funcionais, que obtém e disponibilizam informações, protegendo-as a fim de evitar alterações acidentais ou propositais, conforme [FIS90].

2.6.2 FERRAMENTAS DE GERENCIAMENTO DE PROJETOS

Os gerentes de projetos contam com um amplo conjunto de ferramentas de gerência que quando utilizadas, trazem significativas melhorias na qualidade do trabalho realizado no esforço de desenvolvimento de software, tanto para pequenos quanto grandes projetos.

Segundo [PRE95], ao usar um conjunto selecionado de ferramentas CASE, o gerente de projetos pode gerar estimativas de esforço, custo e duração de um projeto de software, definir uma estrutura de divisão de trabalho, planejar uma programação viável de projeto e acompanhar projetos em base contínua. Com a utilização de ferramentas de gerência, pode-se obter métricas que demonstrarão índices de produtividade de desenvolvimento de software e os níveis de qualidade do produto.

Encontram-se nessa categoria, as ferramentas de planejamento de projetos das quais fazem parte as ferramentas de esforço e estimativa de custos que possibilitam ao gerente de projetos estimar o tamanho do mesmo, utilizando-se para isso de métricas. Algumas dessas ferramentas permitem que o gerente de projetos estabeleça diferentes prazos finais do projeto, a fim de analisar seu impacto sobre o custo global. Também encontram-se nessa categoria, ferramentas de programação de projetos que permitem que o gerente defina as tarefas do projeto, estabeleça uma rede entre as mesmas e represente suas interdependências.

Também fazem parte dessa categoria, as ferramentas de rastreamento de requisitos que tem por finalidade, corrigir o problema da diferença entre o sistema entregue e os requisitos originalmente definidos pelo usuário. Segundo [PRE95], o objetivo das ferramentas de rastreamento de requisitos é oferecer uma abordagem sistemática ao isolamento dos

requisitos que se inicia com a especificação do cliente. Ferramentas típicas dessa categoria combinam a avaliação interativa de textos de usuários com um sistema gerenciador de banco de dados que tem por função armazenar e categorizar cada requisito do sistema a partir da especificação original.

Nessa categoria, encontram-se ainda as ferramentas de métricas de gerenciamento que possibilitam ao gerente de projetos melhor gerir seu trabalho, bem como melhorar a qualidade do software que é produzido. As atuais ferramentas CASE de métricas de gerenciamento estão concentradas nas características de processo de desenvolvimento e de produto de software. Muitas dessas ferramentas armazenam um banco de dados com métricas de qualidade e produtividade de processo e produto a fim de avaliar e comparar suas próprias medidas, com o objetivo principal de melhorar seu desempenho através de estratégias sugeridas pela própria ferramenta.

2.6.3 FERRAMENTAS DE SUPORTE

Nessa categoria, incluem-se ferramentas de aplicação e de sistemas que têm por objetivo complementar o processo de Engenharia de Software.

Dentre essas, existem as ferramentas destinadas à produção de documentos e editoração eletrônica. Devido ao fato das organizações gastarem muito tempo na elaboração da documentação de software, podendo chegar até a 30% de todo o seu esforço de desenvolvimento, as ferramentas CASE destinadas à documentação são essenciais no incremento da produtividade, afirma [PRE95].

Incluem-se também nessa categoria, as ferramentas de software básico que tem por finalidade disponibilizar ao ambiente CASE, software básico de alta qualidade, de rede, correio eletrônico e outras características e possibilidades de comunicação, provendo serviços de portabilidade entre os diversos ambientes operacionais em que as ferramentas CASE operam.

Também fazem parte dessa categoria, as ferramentas de garantia da qualidade que segundo [PRE95], a maioria delas afirmam concentrar-se na garantia de qualidade, sendo de

fato, ferramentas de métrica que fazem a auditoria do código fonte para determinar o cumprimento de padrões de linguagem.

Tem-se ainda as ferramentas de gerenciamento de bancos de dados que tem como objetivo, estabelecer a base para a construção do banco de dados CASE, também chamado de repositório ou enciclopédia. A utilização de ferramentas de gerenciamento de configuração e de bancos de dados, constituem importante passo na criação de uma biblioteca que tem como objetivo primordial, a reutilização de componentes de software [PRE95].

2.6.4 FERRAMENTAS DE ANÁLISE E PROJETO

As ferramentas de análise e projeto possibilitam a criação do modelo de software que será desenvolvido, bem como provêm a capacidade de avaliar a qualidade do próprio modelo. Modelo esse que contém a representação do fluxo de controle dos dados, conteúdo dos dados, representações de processos, especificações diversas de controles e demais representações necessárias à modelagem do software.

Conforme [FIS90], as ferramentas para análise e projeto geralmente implementam vários tipos de diagramas que são úteis na representação do fluxo de informação no sistema.

Encontram-se nessa categoria, as ferramentas destinadas à prototipação e a simulação que provêm ao desenvolvedor, a capacidade de verificar o comportamento de um sistema em tempo real, antes mesmo de sua construção, possibilitando a execução de simulações do sistema para apreciação do usuário final.

Existem ainda nessa categoria, as ferramentas de projeto e desenvolvimento de interfaces. Essas podem ser comparadas a um conjunto de ferramentas de componentes de programa do tipo, ícones, menus, botões e outros. Segundo [PRE95], os sistemas de desenvolvimento de interface com o usuário, combinam ferramentas CASE individuais para a interação homem-computador com uma biblioteca de componentes de programa que possibilita ao desenvolvedor construir a interface o mais rapidamente possível e de acordo com padrões aceitáveis pelo usuário final.

2.6.5 FERRAMENTAS DE PROGRAMAÇÃO

Pertencem a categoria de ferramentas de programação, editores, depuradores e compiladores que estão disponíveis a fim de apoiar a maioria das linguagens de programação usuais. Linguagens de 4ª geração – 4GL, geradores de aplicação, linguagens de consulta a bancos de dados – *QUERY*, ambientes de programação orientados a objeto também se situam nessa categoria.

As ferramentas de codificação de quarta geração oferecem dispositivos de acesso a bancos de dados e facilitam aos desenvolvedores representar suas aplicações num nível de abstração mais elevado. Sistemas de consultas a bancos de dados, geradores de código e linguagens de quarta geração mudaram o modo pelo qual os sistemas têm sido desenvolvidos, pois uma das metas dessas ferramentas é eliminar o encargo tedioso de acesso a bancos de dados, substituindo-o por uma quantidade muito menor de código gerado por 4GL de mais alto nível, projetadas especificamente para esse acesso, conforme [FIS90].

Conforme destaca [FIS90], a geração automática de código que constitui um dos objetivos fundamentais das ferramentas CASE, é a capacidade de gerar automaticamente um software executável diretamente a partir de uma especificação de projeto. Segundo [PRE95], as ferramentas CASE baseadas nas 4GL não apenas convertem uma descrição de sistema em código de programa operacional, mas também ajudam a verificar se a especificação do sistema desenvolvido está de acordo com os requisitos originais do usuário.

2.6.6 FERRAMENTAS DE INTEGRAÇÃO E TESTE

Dentre as ferramentas de integração e teste de software, estão incluídas as categorias de ferramentas de aquisição de dados, medição (análise) estática e dinâmica e gerenciamento de teste.

As ferramentas de aquisição de dados são responsáveis pela extração dos dados a serem usados durante o processo de teste de software.

As ferramentas de análise estática apenas processam a análise do código fonte do software, porém não executam os casos selecionados para teste. Sua função principal é selecionar casos para a realização dos testes.

As ferramentas de análise dinâmica destinam-se a simular as atribuições e funções do hardware ou outros equipamentos externos que integram o processo. Realizam seus testes interagindo com o software em plena execução a fim de cobrir caminhos, testando valores pressupostos de variáveis e provendo de dados, os diversos fluxos de dados previstos para o sistema. As ferramentas de análise dinâmica podem mudar o software testado, introduzindo instruções extras para analisar os resultados do teste. Ferramentas de análise dinâmica podem ser utilizadas conjuntamente com ferramentas de análise estática, sendo que esta, é usada para derivar os casos de teste a serem monitorados pela ferramenta de análise dinâmica, conforme orienta [PRE95].

As ferramentas de gerenciamento de teste têm a função de auxiliar nos processos de planejamento, desenvolvimento e controle dos testes a serem realizados.

2.6.7 FERRAMENTAS DE PROTOTIPAÇÃO

Um dos pontos principais da Engenharia de Software, sendo amplamente utilizado, é a prototipação de software. Todas as ferramentas incluídas nessa categoria situam-se em algum ponto da escala do espectro de implementação que vai de 0% a 100%. Segundo [PRE95], utilizando-se de ferramentas de prototipação, é possível criar imagens reais do sistema para serem analisadas pelo usuário final, a fim de verificar se a função e o comportamento do sistema atendem aos requisitos inicialmente expostos. Algumas ferramentas de prototipação geram imagens estáticas em tela e são ditas ferramentas que geram protótipos em papel para figurar a situação. Outras, geram imagens dinâmicas inclusive gerando código fonte para que as telas possam executar. Muitas das ferramentas que utilizam linguagens de quarta geração possuem características de prototipação.

2.6.8 FERRAMENTAS DE MANUTENÇÃO

A manutenção de software absorve muito tempo do esforço relacionado ao desenvolvimento de sistemas, sendo responsável, muitas vezes, pelo consumo de até 70% do

tempo total de um projeto. As ferramentas de manutenção podem ser subdivididas nas seguintes categorias:

- a) ferramentas de engenharia reversa para especificação: tem o código fonte como entrada e gera modelos gráficos de análise e projeto estruturados e outras informações pertinentes ao projeto;
- b) ferramentas de análise e reestruturação de código: tem por função, analisar a sintaxe do programa, gerar um gráfico de fluxo de controle e gerar automaticamente um novo programa com código altamente estruturado;
- c) ferramentas de reengenharia de sistemas: são utilizadas com o objetivo de modificar sistemas de bancos de dados *on-line*.

As ferramentas de engenharia reversa tem um amplo potencial de uso devido ao fato de que aproximadamente 60% de todo esforço em atividades de software são gastos em manutenção ou modificação de sistemas existentes, conforme [JOA93]. Analisando esse percentual altamente elevado, verifica-se a necessidade de adoção de ferramentas de engenharia reversa por parte das organizações, a fim de diminuir drasticamente seus esforços e gastos na manutenção e modificação de seus sistemas. Segundo [SIL98], ser capaz de fazer engenharia reversa permite visualizar melhor a estrutura da aplicação, as dependências entre os módulos e o impacto de mudanças.

As ferramentas de reengenharia existentes costumam ser classificadas em duas subcategorias: ferramentas de reestruturação de código e ferramentas de reengenharia de dados. As ferramentas de reestruturação de código tem como entrada código não estruturado, executam a análise da engenharia reversa, reestruturando o código a fim de que ele represente conceitos da programação estruturada ou orientada a objetos. As ferramentas de reengenharia de dados tem por objetivo analisar as definições de dados, bancos de dados de uma linguagem de programação ou ainda, linguagens de descrição de bancos de dados e produzir uma notação gráfica para o engenheiro de software a partir da tradução da descrição de dados do programa.

Conforme [PRE95], utilizando-se dessas ferramentas, é possível ao engenheiro de software modificar a estrutura lógica do banco de dados, normalizar os arquivos resultantes e depois gerar novamente o projeto físico do banco de dados.

2.6.9 FERRAMENTAS DE ESTRUTURA

A tendência para as ferramentas CASE é que trabalhem em ambientes integrados. Nesse contexto, as ferramentas de estrutura exercem um papel fundamental, pois oferecem gerenciamento de bancos de dados, gerenciamento de configuração e capacidade de integração de várias ferramentas CASE num mesmo ambiente a fim de que todas trabalhem harmonicamente para alcançar um objetivo comum, que é abranger todas as etapas do ciclo de vida do desenvolvimento de software.

A maioria das ferramentas de estrutura implementam um banco de dados orientado a objetos com um conjunto de ferramentas internas com o propósito de estabelecer interfaces facilitadoras à integração com ferramentas CASE de outros fornecedores, oferecendo capacidade de gerenciamento de configuração, possibilitando o controle de mudanças em todos os itens de configuração criados por todas as ferramentas CASE que estejam integradas à ferramenta de estrutura, segundo [PRE95].

2.7 INTRODUZINDO A TECNOLOGIA CASE NA ORGANIZAÇÃO

Normalmente, introduzir uma nova tecnologia numa organização é um processo difícil, independente dos benefícios e redução de custos que essa tecnologia trará. Esse princípio também se aplica à Engenharia de Software Auxiliada por Computador. As pessoas e as organizações são resistentes às mudanças, principalmente no caso de CASE que exige uma nova forma de desenvolver software. Apesar de todos os empecilhos apresentados, é fundamental que a organização saiba o quanto irá ganhar com a adoção da tecnologia CASE.

Segundo [FIS90], as organizações costumam se apegar a preconceitos baseados em experiências passadas. Além disso, se as políticas e procedimentos correntes estão funcionando, a gerência tende a responder negativamente. A partir dessa situação, deve surgir a figura do defensor da tecnologia CASE para convencer a gerência de que através do uso de CASE, obter-se-á incremento de produtividade, melhoria da qualidade do software desenvolvido e maior facilidade em manutenções futuras.

O defensor da CASE deve dispor de um conjunto de estratégias para convencer a gerência em investir na tecnologia, dentre as quais pode-se destacar a apresentação de

organizações em que a introdução de CASE deu certo, artigos em jornais e revistas especializadas falando acerca de casos de sucesso com CASE, visitas de fornecedores da tecnologia, com a finalidade de apresentar vantagens e facilidades que trarão ao desenvolvimento de software e talvez a mais difícil, convencer a gerência a investir em CASE para apreciação e testes dentro da própria organização.

Muitos são os benefícios alcançados através da utilização de ferramentas CASE por parte das organizações durante o processo de desenvolvimento de software. Porém, muitos desses benefícios poderão não ser obtidos se não houver uma ampla e correta preparação da organização e de seu pessoal de desenvolvimento, na aceitação e utilização da tecnologia de ferramentas CASE.

Ferramentas CASE não é apenas um conjunto de ferramentas que tem por objetivo automatizar as fases do ciclo de desenvolvimento de software. É muito mais que isso, pois além das vantagens técnicas obtidas com seu uso, a CASE exige na maioria dos casos, uma mudança completa de atitudes quanto ao desenvolvimento de software devido ao fato de que as organizações nem sempre estão familiarizadas à utilização de metodologias em suas atividades de produção de software.

Questões importantes sempre surgem: como implantar CASE na organização e obter todos os benefícios que esta tecnologia proporciona ? Quais as etapas que devem ser seguidas a fim de que não seja traumática para a organização e para as pessoas que nela trabalham, a implantação de CASE ?

Segundo [FOU94], existe um guia composto por seis etapas, cuja finalidade é construir uma estratégia de implementação especialmente adaptada às necessidades únicas de cada organização:

- a) avaliar o ambiente corporativo de informática:
 - levantar pontos fortes e fracos;
 - verificar padrões de desenvolvimento, de configuração de hardware, software e ambiente de rede;
 - verificar tipos de ferramentas e métodos de Engenharia de Software utilizados atualmente bem como sua efetividade;
 - verificar procedimentos de garantia de qualidade de sistemas;

- b) desenvolver um conjunto significativo de requisitos:
 - descrever os requisitos da área de informática da organização considerando, além dos critérios de software, a configuração de hardware adotada;
 - verificar quais áreas de informática apresentam facilidade de aumento de produtividade e qualidade e verificar quais são as necessidades que devem ser atendidas com maior urgência;
 - analisar a escolha das ferramentas a adquirir, levando-se em conta os investimentos anteriores em tecnologia e treinamento da equipe de desenvolvimento;
 - definir a melhor forma de introdução de ferramentas CASE na organização: adotar uma ferramenta que automatiza uma fase individual do processo, ou adotar um ambiente totalmente integrado, que cobre todas as fases do ciclo de desenvolvimento de software;
- c) desenvolver um plano de transição adaptado à organização:
 - elaborar um plano de ações que referencie a forma como a organização irá satisfazer as necessidades e prioridades na implementação de CASE;
 - elaborar o plano de transição a ser seguido, para que os objetivos da organização sejam alcançados, considerando o estado atual da tecnologia na organização, as necessidades específicas da área de informática, o orçamento alocado para o processo de adoção de CASE, o comprometimento da gerência e o prazo de implementação proposto;
- d) familiarizar-se com a tecnologia CASE:
 - verificar se a tecnologia apresentada é comprovadamente eficaz ou não, através da participação em seminários, eventos ou simpósios especializados onde os fornecedores de CASE apresentam seus produtos;
 - requerer dos fornecedores a apresentação de provas que atestem de que o produto oferecido é melhor que as ferramentas dos concorrentes;
 - entrar em contato com outras organizações que adotaram a tecnologia CASE, a fim de constatar a veracidade das informações repassadas pelos fornecedores;
- e) formular critérios de seleção de ferramentas CASE específicas:
 - adquirir apenas ferramentas compatíveis com o ambiente existente de

- hardware e software da organização;
 - adquirir ferramentas CASE que suportem a metodologia e os padrões de desenvolvimento de sistemas seguidos pela organização;
 - adquirir ferramentas que realmente atendam as necessidades da organização;
- f) implementar a tecnologia CASE:
- alocar recursos e meios necessários para um amplo treinamento dos usuários das ferramentas quanto ao uso da metodologia que será utilizada para desenvolvimento de software, bem como quanto ao uso da própria ferramenta;
 - estabelecer acompanhamentos e diretrizes de uso de CASE para que sua adoção proporcione o incremento de produtividade e qualidade esperados da tecnologia;
 - realizar apresentações sobre conceitos, questões e benefícios relacionados às ferramentas CASE a fim de que o usuário dessa tecnologia se torne um componente integral do processo de automação da Engenharia de Software;
 - implementar a tecnologia CASE na organização através de projetos piloto, pois a medida que esse projetos são concluídos com sucesso, a adoção de CASE será justificada e gradualmente se espalhará por toda a área de informática da organização.

Deve-se buscar através da implementação desses seis passos, uma perfeita assimilação da tecnologia CASE, a fim de que a organização aproveite ao máximo todos os benefícios proporcionados por ela.

Não há dúvidas quanto aos ganhos relacionados à implantação de CASE na organização, principalmente no tocante ao incremento de produtividade da equipe de desenvolvimento e aumento da qualidade dos produtos gerados. Com isso, produz-se software de mais fácil manutenção e conseqüentemente, produtos em melhores condições de competir no mercado.

2.8 O FUTURO DAS FERRAMENTAS CASE

As ferramentas CASE por estarem diretamente associadas ao uso de metodologias de desenvolvimento de software, seguirão as mudanças e avanços implementados pelas mesmas, considerando também a possibilidade de desenvolvimento de novas metodologias. Segundo [FIS90], as ferramentas CASE se tornarão altamente personalizáveis a fim de incorporarem as novas metodologias inclusive as definidas pelo próprio usuário.

O aumento do uso de ferramentas CASE é um ponto dado como certo para os próximos anos devido ao fato de uma maior tomada de consciência por parte dos administradores das organizações, na questão referente à importância estratégica do uso de tecnologias para o desenvolvimento de sistemas e o fato de que as equipes de desenvolvimento têm cada vez mais a certeza de que devem utilizar metodologias associadas a ferramentas CASE para que incrementem sua produção de software e se tornem mais competitivas.

As ferramentas CASE do futuro estarão providas de todos os recursos necessários para a geração automática do software a partir da especificação do projeto e da análise dos requisitos, pois as ferramentas CASE devem proporcionar as mais poderosas capacidades de geração de código, segundo [MAR95]. Através da liberação do desenvolvedor da demorada tarefa que é a codificação, ele terá mais tempo para os processos iniciais do ciclo de desenvolvimento de software que determinam a qualidade do software com o completo atendimento dos requisitos do usuário. Raras serão as necessidades de codificação manual, pois conforme [FIS90], as exceções se darão pela necessidade de codificação personalizada para aplicativos com características exclusivas. Importante ressaltar a necessidade da implementação de facilidades de incorporação desse trabalho manual no laço de especificação-projeto-implementação sem que se perca a sincronia com o projeto original, não comprometendo sua evolução.

Um fator esperado para o futuro, é um amplo uso de ferramentas de engenharia reversa, pois conforme [JOA93], a inabilidade para automatizar a manutenção de milhões de linhas de código existentes constitui uma séria falha em tecnologia à maioria das empresas. Devido ao fato de muitos sistemas terem sido construídos sem o uso de uma metodologia adequada e padrões de desenvolvimento, a utilização de ferramentas de engenharia reversa

terá um campo crescente de aplicação, pois as organizações buscarão cada vez mais diminuir seus custos de produção de software e a manutenção de sistemas. Segundo [MAR95], utilizando-se de ferramentas I-CASE baseadas em repositórios, a manutenção do futuro consistirá em se adicionar características aos sistemas ou mudá-las de forma evolutiva.

Outro ponto importante é que continuará em escala ascendente no futuro, é a implementação de software reusável, devido ao alto custo do software em relação ao hardware. As equipes de desenvolvimento buscarão com empenho desenvolver módulos de software que possam ser utilizados em mais de um projeto a fim de baixar os custos do mesmo, fazendo da reusabilidade uma resposta importante aos problemas de produtividade, afirma [JOA93]. Para isso utilizarão ferramentas CASE que farão da reusabilidade uma estratégia prática de construção de sistemas. Os módulos de software serão armazenados em bibliotecas de componentes reutilizáveis. Para ter-se um incremento no uso dessas bibliotecas, as ferramentas devem prover a capacidade de armazenamento de funções de programa desenvolvidas manualmente para serem aplicadas em softwares específicos que necessitem de módulos com características próprias, afirma [FIS90]. O software do futuro será construído montando-se e customizando-se componentes existentes com ferramentas baseadas em repositórios. Com o objetivo de desenvolver componentes de software reusáveis deve-se buscar a implementação de algumas propriedades desejáveis conforme descreve [MAR95]:

- a) as equipes de desenvolvimento devem empregar em seu ambiente CASE integrado (I-CASE) as técnicas de projeto orientadas a objeto a fim de maximizar a reusabilidade;
- b) independência de componentes e ter um comportamento previsível;
- c) independência de linguagem de programação;
- d) interfaces claras, simples e precisas;
- e) auto-organização, pois os componentes devem saber de quais outros componentes eles precisam.

Uma outra característica das ferramentas CASE do futuro relacionada à reusabilidade de software é o fato de que a indústria CASE deverá incluir bibliotecas de componentes reusáveis em suas ferramentas que serão armazenadas no repositório.

Busca-se para o futuro o estabelecimento de padrões de repositório com o objetivo de integrar as ferramentas CASE, facilitando a aquisição dessas ferramentas a partir de diversos

fornecedores com a possibilidade real de integração entre si. É essencial para o futuro da CASE a existência de padrões abertos para o repositório e suas interfaces com suas ferramentas. Segundo [MAR95], padrões abertos de repositório devem incorporar o seguinte:

- a) um repositório com conteúdo definido em termos dos tipos de objetos que ele armazena;
- b) serviços de repositório para verificar a integridade dos dados armazenados;
- c) controle de versão dos objetos armazenados;
- d) formatos padrões para solicitações e respostas do objeto;
- e) uma interface gráfica padrão que faça as ferramentas similares entre si e fáceis de usar;
- f) uso completo de padrões existentes de sistemas abertos.

Para [FIS90], constitui a restrição tecnológica fundamental o desenvolvimento de uma estrutura central de banco de dados ou repositório que possa integrar e sintetizar os componentes de ferramentas distintas. A necessidade da adoção de padrões de repositório por parte da indústria de software é justificada pois ele não significa apenas um dicionário de dados, pois além disso, ele armazena outros tipos de informações do sistema como a especificação dos requisitos do usuário, relacionamentos entre os componentes e as regras para uso de componentes e formas para seu processamento.

As ferramentas destinadas a implementação de interfaces com o usuário deverão tornar-se mais orientadas a objetos, devido ao fato de que a programação orientada a objetos trouxe nos últimos anos, um novo estilo de programação que se concentra em estruturas de dados denominadas objetos em vez dos algoritmos que as manipulam. Isso possibilita ao desenvolvedor dedicar-se mais aos objetos com os quais o usuário terá contato pois para ele, o usuário, o que importa é o ambiente que ele interage e não as estruturas internas que estão por trás desse ambiente.

Enfim, o futuro reserva para a tecnologia CASE, ambientes integrados, a partir de ferramentas de diversos fornecedores que terão sua intercomunicação facilitada e garantida pela implementação de repositórios padronizados. Esses repositórios além de todas as informações referentes ao projetos e à organização, conterão bibliotecas de componentes reutilizáveis para serem usados no maior número possível de softwares a serem desenvolvidos. O crescente uso de software reusável proporcionará às organizações os níveis

de produtividade almejados no desenvolvimento de software, com conseqüente melhoria da qualidade dos sistemas a custos cada vez mais competitivos.

2.9 O MERCADO DE FERRAMENTAS CASE

O mercado de ferramentas CASE possui um grande potencial de crescimento nos tempos atuais e também para os próximos anos. Isso se deve à necessidade das organizações automatizarem os processos de criação e manutenção de software, para que o mesmo seja um produto de qualidade atendendo aos requisitos do usuário a um custo cada vez menor. Nesse contexto, a utilização dessas ferramentas constitui parte integrante e indispensável do ambiente de desenvolvimento de software, pois tais vantagens são proporcionadas pelo uso da CASE.

A obra de [MAY98] apresenta uma pesquisa acerca do mercado brasileiro quanto ao desenvolvimento, utilização e comercialização de software. Nesta obra, uma parte importante é a que se referente à utilização de ferramentas de desenvolvimento nas organizações que desenvolvem software, seja para uso interno, seja com o objetivo de venda ou prestação de serviços a terceiros. Algumas ferramentas CASE são citadas na pesquisa por fazerem parte do conjunto de ferramentas do desenvolvedor de software, como por exemplo: *Designer 2000*, *ErWin*, *System Architect* e *Sybase Designer*. As ferramentas citadas, constituem ferramentas CASE típicas, ou seja, ferramentas que são empregadas nas fases de análise, projeto e implementação de software, sendo classificadas, segundo [PRE95], em: ferramentas de planejamento de sistemas comerciais, gerenciamento de projetos, suporte, análise e projeto, programação, integração e teste, prototipação, manutenção e ferramentas de estrutura. Uma constatação a que se pode chegar analisando esta obra, é o baixo índice de utilização dessas ferramentas no desenvolvimento de software. Espera-se que a médio prazo, essa estatística se modifique pois, torna-se cada vez mais necessário baixar custos de desenvolvimento de software, fator esse que pode ser obtido através da utilização de ferramentas CASE, ao proporcionar aumento de produtividade da equipe de desenvolvimento e aumento da qualidade dos softwares produzidos. A tendência natural para um futuro próximo, é que a estatística se torne favorável quanto ao uso de CASE pois, a partir de uma concorrência mais acirrada entre as organizações, a busca por produtividade, qualidade e competitividade, constitui fator prioritário para as mesmas.

2.10 AVALIAÇÃO DE FERRAMENTAS CASE

Um dos pontos fundamentais relacionados à tecnologia CASE é a avaliação que se faz das ferramentas disponíveis. Pouco adianta para a organização comprar a primeira ferramenta ou conjunto de ferramentas que encontrar no mercado e que esteja dentro do seu orçamento, se não houver sido realizada anteriormente uma rigorosa avaliação para uma escolha segura e que atenda as reais necessidades da organização.

Avaliar ferramentas CASE é muito mais que simplesmente comparar preços e condições de pagamento. Caso a organização ainda não esteja familiarizada com nenhuma metodologia de desenvolvimento de sistemas, é preciso definir e estudar essa metodologia antes mesmo de comprar ferramentas.

Um ponto importante é a necessidade de treinamento para a utilização da ferramenta, bem como suporte técnico após a mesma ter sido implantada na organização, pelo menos durante o desenvolvimento dos primeiros projetos com CASE. Segundo [FIS90], após a definição das necessidades da organização, algumas questões importantes devem ser respondidas a fim de que os riscos de adoção de CASE sejam minimizados:

- a) a ferramenta tem um escopo bem definido ? Após definidos os requisitos da organização, deve-se proceder à escolha de ferramentas que os satisfaçam, não se deixando levar pelo apelo dos fornecedores e sim, pelo atendimento integral dos requisitos definidos;
- b) a ferramenta poderá ser personalizada se necessário ? Algumas organizações desenvolveram extensões das metodologias padrão existentes ou desenvolveram metodologias originais e necessitam acrescentar seus próprios símbolos e regras para obter da ferramenta a automação seguindo seus princípios e práticas de desenvolvimento;
- c) a integração é suficiente para auxiliar as tarefas de projeto e desenvolvimento de sistemas ? A ferramenta precisa oferecer uma vantagem maior do que as resultantes dos métodos atualmente empregados;
- d) a ferramenta gera software automaticamente a partir da especificação do projeto ? A geração automática do software é uma característica amplamente procurada nas ferramentas CASE atuais, com a garantia de que o software

gerado seja isento de erros e de fácil manutenção, também realizada pela ferramenta.

A avaliação de ferramentas CASE é um processo geralmente demorado e difícil de ser implementado, pois muitas vezes precisa-se desenvolver um software completo para que se obtenha uma análise abrangente e real acerca da ferramenta a fim de verificar se ela atende a todos os requisitos propostos pela organização.

Uma forma bastante comum para o processo de avaliação é a utilização de questionários que buscam abranger todas as características de ferramentas CASE. De acordo com as necessidades da organização, algumas características tornam-se mais importantes que outras. Todas as perguntas formuladas visam esclarecer a abrangência, características e pontos importantes acerca de ferramentas CASE. Seguindo a abordagem de [FIS90], pondera-se alguns pontos relevantes sobre ferramentas CASE, conforme [FOU94] questiona:

- a) existem padrões que regulem a tecnologia CASE ? É fundamental que a ferramenta adote padrões estabelecidos pela indústria de software a fim de facilitar a integração entre ferramentas de diversos fornecedores quando essa estratégia for a mais significativa à organização;
- b) a ferramenta CASE é amigável para o usuário ? É necessário questionar alguns aspectos como: a possibilidade de utilização de janelas, funções de mouse, recursos gráficos de alta resolução, teclas de função, facilidade de interpretação de mensagens de erros e consultas ao repositório central;
- c) qual o futuro proposto pelo fornecedor do produto CASE ? Verificar a capacidade de evolução do fornecedor, seus investimentos em pesquisa e desenvolvimento e sua tendência e intenções declaradas de adotar futuros padrões determinados pela indústria de software CASE;
- d) a ferramenta CASE possibilita reutilização ? É fundamental que a ferramenta disponibilize recursos para a reutilização de partes de análise e projeto, bem como a utilização de partes de código reutilizável de outros projetos que estão armazenados no repositório CASE.

Um ponto fundamental quando se avaliam ferramentas CASE é a análise da relação custo/benefício que a sua adoção trará para a organização. Pontos chave como a possibilidade de geração automática do software e a capacidade de reutilização tanto de partes de projeto

como partes de código implementado, evidenciam os objetivos principais da adoção de CASE por parte da organização que são o incremento da produtividade da equipe de desenvolvimento e a geração de software de qualidade que facilite sua manutenção.

A medida que se busca especialização e estudos sobre formas de avaliar ferramentas CASE, mais descobrem-se novos critérios e questões relevantes que devem ser analisados. Conforme apresentado na obra de [JOA93], alguns aspectos importantes sobre a avaliação de ferramentas CASE devem ser verificados, tais como:

- a) capacidade de gerar protótipos: uma importante característica que agiliza o desenvolvimento de software, bem como sua interação com o usuário final, é a capacidade de se desenvolver protótipos que tem por finalidade verificar se os requisitos do usuário estão sendo ou não atendidos;
- b) capacidade de documentar o projeto: a documentação do projeto e da implementação do software constituem fator básico na avaliação de ferramentas CASE devido a sua importância para o registro das atividades desenvolvidas e facilitação do trabalho em futuras manutenções do projeto e do software;
- c) permitir distribuição de atividades: a ferramenta deve permitir que diversos profissionais atuem em diferentes áreas tanto de projeto quanto de implementação e facilitar sua completa integração e controlar a execução dessas mesmas atividades;
- d) facilidade de manter o projeto: não é suficiente facilitar a manutenção do software, mas principalmente manutenção no projeto que constitui a base para a implementação do software.

Uma outra abordagem para a avaliação e seleção de ferramentas CASE é apresentada no trabalho de [BLA94], que propõe o processo de avaliação e seleção de ferramentas CASE em três fases, a seguir:

- a) apresentação de ferramentas CASE: nessa primeira fase, três perguntas devem ser debatidas: Quais ferramentas CASE estão disponíveis ?; Quais pacotes de ferramentas CASE poderiam ser seriamente considerados para serem avaliados em detalhes ?; e, há uma única ferramenta que pode ser utilizada ou poderia se usada uma combinação de ferramentas ? Após essas considerações, deve-se elaborar uma pequena lista de ferramentas candidatas para avaliação.

Inicialmente, a equipe de projeto deve identificar a metodologia de desenvolvimento de sistemas utilizada pela organização e verificar quais ferramentas atendem ou não a essa metodologia. Também devem considerar a configuração de hardware e software utilizada na organização, para essa confrontação. Deve-se elaborar uma lista de critérios de avaliação, baseada em requisitos funcionais não comumente apresentados pelas ferramentas, pois essas podem ter capacidades e características únicas que podem prontamente distinguir uma ferramenta de outra. Identificados esses requisitos funcionais, deve-se confrontá-los com os requisitos únicos da organização a fim de identificar as ferramentas que atendem a um maior número deles, para serem melhor avaliadas na próxima fase. Seguindo a apresentação de critérios de ferramentas CASE, devem ser considerados: requisitos técnicos, como: sistemas operacionais e linguagens de programação suportados, periféricos, memória necessária e capacidade de comunicação de dados; requisitos funcionais, como: suporte para as fases do ciclo de vida do desenvolvimento de software, características de integração e padrão de interface de usuário; documentação, que deve ser completa, detalhada e de fácil compreensão; treinamento adequado; e informações acerca do fornecedor, tais como: sua habilidade de fornecer treinamento, consultoria e suporte na instalação e manutenção da ferramenta, bem como sua estabilidade financeira perante o mercado. Após a análise desses critérios e requisitos, deve-se elaborar uma lista contendo as ferramentas que têm reais condições de atender aos requisitos da organização, para serem melhor avaliadas na fase seguinte;

- b) avaliação de ferramentas CASE: essa segunda fase irá trabalhar com duas ou três ferramentas selecionadas na fase anterior. Seu objetivo é analisar mais detalhadamente as finalistas e selecionar uma ferramenta ou um conjunto delas que melhor atende às necessidades da organização. Dentre as tarefas dessa fase, destacam-se: definir critérios de avaliação detalhados, obter informações mais específicas acerca das ferramentas e avaliar as finalistas e selecionar a ferramenta ou conjunto delas como sendo a melhor alternativa para a organização. Para a definição dos critérios de avaliação mais específicos, deve-se expandir em mais detalhes os requisitos técnicos, funcionais, de documentação, de treinamento e

informações do fornecedor, analisados na fase de apresentação de ferramentas CASE. Para a avaliação, deve-se buscar pontos fortes e fracos de cada pacote de CASE. Nessa fase, dá-se ênfase à análise do que a ferramenta não possui, como exemplo, funções inexistentes e como essas poderiam ser implementadas. Pode-se atribuir pesos a cada requisito ou critério analisado para facilitar a comparação posterior entre as ferramentas analisadas. Deve-se tomar cuidado, pois a maior pontuação pode não refletir a melhor ferramenta, pois podem não representar fatores como a aparência das telas ou a facilidade de utilização da ferramenta. A pontuação também pode não indicar o nível de habilidades necessárias para a implementação da ferramenta;

- c) confirmação da seleção da ferramenta CASE: após a escolha da ferramenta, é necessário confirmar tal escolha. Isso se processa através do desenvolvimento de algumas aplicações específicas baseadas na ferramenta selecionada. A razão principal para essa fase, é garantir que a ferramenta selecionada, atende efetivamente os requisitos da organização, podendo a decisão ser reconsiderada, caso necessário. Para que haja uma maior satisfação por parte dos usuários que utilizarão a ferramenta, é importante demonstrar aplicações completas ou partes de sistemas desenvolvidas com o uso da mesma. A apresentação dessas aplicações protótipo pode demonstrar grandes benefícios antes da finalização da decisão de seleção da ferramenta CASE. Nessa fase pode ser necessário a alteração de requisitos funcionais da ferramenta para poder analisar sob uma nova ótica e comprovar a escolha. Pode ser mais prudente que a decisão final de usar uma ferramenta em particular seja evitada até que o projeto de uma aplicação, utilizando o potencial da ferramenta esteja adiantado, num ponto onde a satisfação pelo uso da ferramenta esteja garantida.

Segundo [BLA94], a presente metodologia proposta para avaliação e seleção, visa reduzir os riscos associados a ferramentas CASE e facilita a obtenção de sucesso no desenvolvimento específico de aplicações utilizando-se a tecnologia CASE, que tem dentre seus objetivos, reduzir o número de pessoas envolvidas no processo de desenvolvimento e manutenção de sistemas, bem como reduzir o custo desses mesmos sistemas.

O processo de avaliação de ferramentas CASE deve ser bem elaborado por parte da organização. É aconselhável e necessário a formação de uma equipe de avaliação composta por pessoal da área de planejamento, departamento financeiro e profissionais envolvidos no desenvolvimento de software. Deseja-se a partir dessa formação, obter diversos pontos de vista acerca da avaliação e adoção de ferramentas CASE na organização. Todos os membros da equipe tem a colaborar, pois cada um analisa sob a sua ótica e poderá contribuir para o sucesso do processo de avaliação que, ao seu final, deverá selecionar a ferramenta que melhor se adapta e atende às reais necessidades da organização. Deve-se avaliar muito bem a ferramenta ou conjunto de ferramentas, antes de sua aquisição e implantação, para que não ocorra prejuízo devido ao alto investimento empenhado, tanto na alocação de recursos financeiros quanto humanos.

3 NORMA ISO/IEC 14102 – AVALIAÇÃO E SELEÇÃO DE FERRAMENTAS CASE

A norma ISO/IEC 14102 – Avaliação e Seleção de Ferramentas CASE encontra-se em estudo na ABNT (Associação Brasileira de Normas Técnicas). Com o propósito de aplicar esta norma no Brasil a fim de servir como referencial para aqueles que desejam avaliar e selecionar ferramentas CASE, a ABNT criou em setembro de 1996, a Comissão de Estudos CE-21:101.05 de Avaliação e Seleção de Ferramentas CASE, subordinada à Comissão Técnica de Engenharia de Software e Portabilidade. Para a elaboração desse trabalho, foi obtida junto a essa Comissão, uma cópia da versão inicial da norma.

A norma ISO/IEC 14102 define uma seqüência de processos e um conjunto estruturado de características de ferramentas CASE, para uso na avaliação técnica e seleção definitiva de uma ferramenta [ISO98]. Esta norma segue a norma NBR 13596/1996 – “Tecnologia de informação – Avaliação de produto de software – Características de qualidade e diretrizes para o seu uso” [NBR96], adotando as características e subcaracterísticas do modelo geral de qualidade de software. Considerando que o produto a ser avaliado e selecionado é uma ferramenta CASE, a norma ISO/IEC 14102 estende as características exclusivas para as mesmas.

Além de uma avaliação técnica responder o quanto a ferramenta CASE atende aos pré-requisitos de seus usuários, responde também a questão do quanto a ferramenta atende às funcionalidades requeridas.

O objetivo do processo de avaliação técnica é apresentar resultados quantitativos nos quais a seleção final possa ser baseada. A mensuração atribui números para as características da ferramenta sendo a principal atividade da avaliação, obter estes valores para o uso na seleção. Os resultados da seleção final devem buscar objetividade, repetibilidade e imparcialidade.

Para serem amplamente aceitos, os processos de avaliação e seleção devem ter utilidade para usuários e fornecedores de ferramentas CASE e para a comunidade em geral. As informações apresentadas na norma devem levar a seleções mais eficientes em termos de

custos e a uma maior uniformidade na descrição de funções e características de ferramentas CASE.

3.1 ESCOPO

A norma, como seu próprio nome diz, trata de avaliação e seleção de ferramentas CASE, cobrindo parcial ou completamente o ciclo de vida da Engenharia de Software. Estabelece processos e atividades a serem aplicadas na avaliação de ferramentas e na seleção da ferramenta mais apropriada dentre várias candidatas. Estes processos são genéricos e as organizações devem adaptá-los de acordo com suas necessidades. Os processos de avaliação e seleção de ferramentas CASE devem ser inseridos no amplo contexto do processo de adoção de tecnologia da organização.

A norma propõe:

- a) orientação na identificação dos requisitos da organização para ferramentas CASE;
- b) orientação no mapeamento destes requisitos para as características das ferramentas CASE a serem avaliadas;
- c) um processo para seleção da ferramenta CASE mais apropriada dentre várias, baseado na mensuração de características definidas.

Os principais usuários da norma são organizações que pretendem adotar ferramentas CASE para suportar seus processos de ciclo de vida de software. Os fornecedores de ferramentas CASE podem também usar a norma para descrever as características das suas ferramentas.

3.2 DEFINIÇÕES

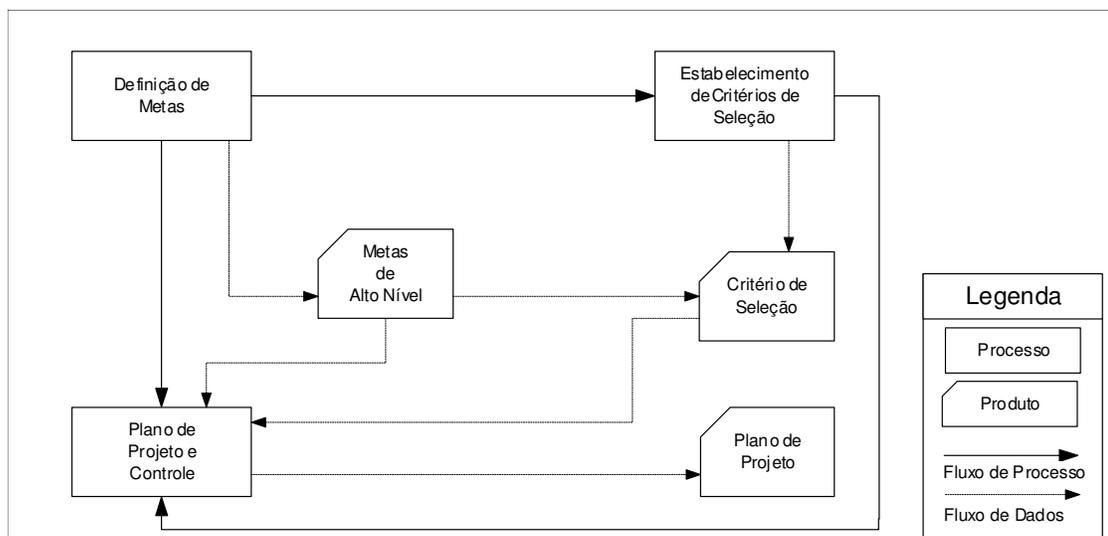
Para um melhor entendimento dos termos usados durante a explanação dos processos constantes da norma, são apresentadas as seguintes definições [NBR96] [NBR97]:

- a) ferramenta CASE: um produto de software que pode auxiliar engenheiros de software através do suporte automatizado para atividades do ciclo de vida de software;

- b) característica: um aspecto de um produto pelo qual ele pode ser descrito e avaliado. Uma característica pode ser refinada em múltiplos níveis de subcaracterísticas que evidenciem a sua capacidade de satisfazer necessidades estabelecidas ou implícitas;
- c) subcaracterísticas atômicas: as características normalmente são subdivididas em subcaracterísticas, que podem ser mais adiante subdivididas em subcaracterísticas de menor nível. No nível mais baixo, quando não houver mais subdivisão, as subcaracterísticas são chamadas de subcaracterísticas atômicas;
- d) medição: aplicação de uma métrica de qualidade de software a um produto de software específico;
- e) métrica: uma escala quantitativa e um método que podem ser usados para determinar o valor que uma subcaracterística recebe em um produto de software específico;
- f) pontuação: mapeamento do valor medido ao nível apropriado de pontuação. Usado para determinar o nível de pontuação obtido pelo software em uma característica de qualidade específica;
- g) nível de pontuação: uma faixa de valores numa escala para permitir que o software seja classificado de acordo com as necessidades explícitas ou implícitas. Níveis de pontuação adequados podem ser atribuídos às diferentes visões de qualidade de usuários, gerentes e equipe de desenvolvimento;
- h) julgamento: aplicação de critérios de julgamento específicos e documentados a um produto, pacote ou módulo de software específico, com o propósito de determinar sua aceitação ou liberação para uso.

3.3 VISÃO GERAL DA AVALIAÇÃO E SELEÇÃO DE FERRAMENTAS CASE

Para a avaliação e seleção de ferramentas CASE, faz-se necessário a implementação de quatro processos que são o processo de iniciação, estruturação, avaliação e seleção. Dentre esses, um processo chave é a estruturação, composta de um conjunto de requisitos sob os quais as ferramentas CASE candidatas serão avaliadas e servirão como base para decisões de seleção. As características de ferramentas CASE definidas no item 3.8 formam a base para a



Fonte: [ISO98] - Norma ISO/IEC 14102 – Avaliação e seleção de ferramentas CASE

Figura 2 : Visão geral do processo de iniciação

3.4.1 DEFINIÇÃO DE METAS, CRITÉRIOS DE SELEÇÃO E PLANEJAMENTO E CONTROLE DO PROJETO

A definição de metas tem por objetivo fornecer a fundamentação para aquisição e o plano geral para avaliação e seleção. Durante esta fase, algumas tarefas devem ser realizadas tais como: desenvolver a fundamentação para aquisição, definir metas e expectativas e estabelecer políticas gerais de aquisição.

Fazem parte dessas tarefas, a revisão do atual processo de desenvolvimento, a análise do estado da arte em tecnologia CASE e suas tendências, a identificação dos impactos de CASE na organização, o desenvolvimento das metas globais como a melhoria de produtividade e da qualidade, a identificação de limites para aquisição de ferramentas, analisar a viabilidade das alternativas de acordo com a disponibilidade da organização, considerações técnicas, especificações de desempenho e recursos.

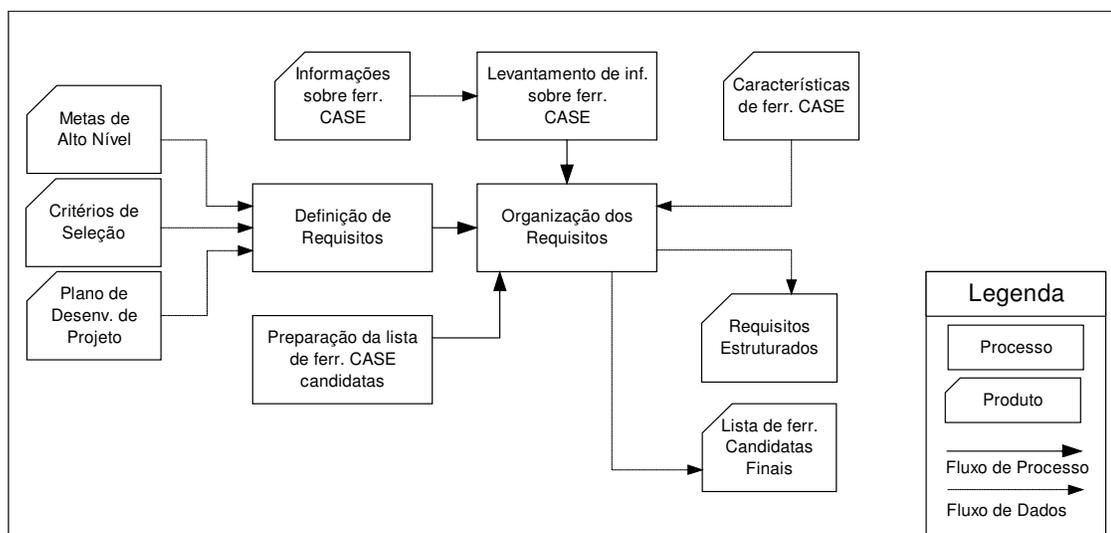
Devem ser estabelecidos critérios de seleção que tem por objetivo fornecer critérios que serão usados no processo de seleção subsequente. Esses critérios tem com base as metas e expectativas desenvolvidas pela organização.

Outro ponto importante é o planejamento e controle do projeto, que visa produzir um plano que inclui informações gerais de planejamento e a implementação de mecanismos de controle baseados nas metas e critérios de seleção que foram estabelecidos para todo o processo de avaliação e seleção. O plano e o mecanismo de controle devem ser desenvolvidos de acordo com o planejamento e processo de controle habituais da organização.

3.5 PROCESSO DE ESTRUTURAÇÃO

O objetivo desse processo é elaborar um conjunto de requisitos estruturados e obter informações das ferramentas. O processo começa com uma atividade de definição de requisitos seguida por duas atividades que ocorrem em paralelo: o levantamento de informações sobre ferramentas CASE existentes e a preparação de uma lista de ferramentas candidatas à avaliação.

A organização dos requisitos das ferramentas baseia-se nas características de ferramentas CASE apresentadas no item 3.8. O processo de estruturação é apresentado na figura 3.



Fonte: [ISO98] - Norma ISO/IEC 14102 – Avaliação e seleção de ferramentas CASE

Figura 3 : Visão geral do processo de estruturação

3.5.1 DEFINIÇÃO DE REQUISITOS

A definição (análise) de requisitos visa transformar as necessidades da organização em estruturas mensuráveis. Um conjunto abrangente de requisitos é necessário para a seleção da ferramenta CASE mais apropriada, e o processo de estruturação facilita o processo de avaliação e sua repetitividade.

Dentro desse processo, faz-se necessário levantar algumas informações acerca da organização tais como: compromisso da organização em prover recursos necessários ao uso de CASE, o atual ambiente de Engenharia de Software, impactos e melhorias esperados das ferramentas CASE bem como a atual política de aquisição de tecnologia da organização.

Outro ponto importante é a identificação de requisitos dos usuários da ferramenta. Esses requisitos devem tratar tanto as funções da ferramenta CASE, quanto seu impacto no ambiente existente. Algumas tarefas devem ser realizadas para a obtenção da lista de requisitos dos usuários da ferramenta tais como: avaliação das necessidades atuais de ferramentas CASE, a seleção da metodologia desejada (exemplo: orientada a processo, orientada a dados, orientada a objetos), identificação das fases do ciclo de vida que serão suportadas, funções requeridas e características e métricas de qualidade pretendidas da ferramenta.

Após o levantamento de todos os requisitos de usuários de CASE, faz-se necessário organizar (estruturar) os mesmos de modo que a avaliação possa prosseguir de forma mais efetiva, através de tarefas como a classificação dos requisitos dos usuários conforme características de ferramentas CASE, detalhadas no item 3.8 e a seleção das características e subcaracterísticas que devem ser avaliadas a fim de atender aos requisitos e identificar os pesos para as mesmas.

3.5.2 LEVANTAMENTO DE INFORMAÇÕES SOBRE FERRAMENTAS CASE

Essa tarefa visa obter informações sobre o estado da arte em ferramentas CASE do momento. Uma busca geral de ferramentas CASE com potencial para avaliação é realizada baseada nos requisitos e critérios de seleção estabelecidos. Para as ferramentas analisadas, são obtidas informações adicionais e mais detalhadas que possam auxiliar a eliminar rapidamente

várias ferramentas, permitindo que se concentre a atenção nas candidatas restantes. Algumas das informações a serem levantadas são:

- a) informações gerais do vendedor;
- b) custo da ferramenta (ex., preço, manutenção, treinamento);
- c) hardware e software necessários para suportar o uso da ferramenta;
- d) treinamento necessário para o uso eficiente da ferramenta;
- e) metodologia e ciclo de vida suportados pela ferramenta;
- f) interfaces da ferramenta com sistemas externos.

3.5.3 IDENTIFICAÇÃO DAS FERRAMENTAS CASE CANDIDATAS FINAIS

Nessa etapa, são identificadas as ferramentas para avaliação, usando-se os resultados obtidos nas duas etapas anteriores. Quando o conjunto das ferramentas candidatas tiver sido identificado, as candidatas finais para seleção podem ser escolhidas. Isto é realizado através das seguintes tarefas:

- a) estabelecer um conjunto de requisitos com alta prioridade que deverão ser atendidos pelas ferramentas CASE;
- b) comparar os requisitos funcionais solicitados pelos usuários com as capacidades funcionais, metodologias suportadas e ambientes das ferramentas CASE;
- c) comparar os requisitos gerenciais com os custos, disponibilidade de treinamento e suporte das ferramentas CASE;
- d) analisar a base de usuários dos vendedores, opiniões dos usuários, tradição de suporte e histórico de mercado.

3.6 PROCESSO DE AVALIAÇÃO

O objetivo do processo de avaliação é produzir relatórios técnicos de avaliação que servirão como base principal para o processo de seleção. A avaliação pode começar quando os requisitos estruturados estiverem definidos e um conjunto representativo das candidatas finais para seleção estiver escolhido. Desenvolve-se então, um plano de avaliação. As atividades de avaliação são executadas e documentadas, resultando num resumo de como cada ferramenta CASE foi mensurada de acordo com os requisitos estruturados. O processo de avaliação é mostrado na figura 4.

3.6.2.1 MENSURAÇÃO

Medições podem ser baseadas em informações obtidas pelo exame da própria ferramenta CASE, ou informações acerca desta, através dos seguintes tipos de tarefas:

- a) entrevistando usuários do software;
- b) observando demonstrações e entrevistando demonstradores;
- c) executando casos de teste;
- d) aplicando em projetos de teste.

Os valores de medição podem ser binários (quantificáveis), ou textuais. Existem características de ferramenta objetivas que são aquelas que permitem testes ou métricas independentes e repetíveis e características subjetivas que são aquelas para as quais não existem testes ou métricas independentes e repetíveis (ex. adaptação da interface do usuário à sua cultura). Os resultados da avaliação devem ser registrados de uma maneira quantificada, quando possível, junto com justificativas textuais, onde aplicável.

3.6.2.2 PONTUAÇÃO

Na tarefa de pontuação, cada valor mensurado é classificado contra uma escala de valores definidos no plano da avaliação. Os níveis de pontuação são gerados ou calculados de acordo com algoritmos previamente definidos.

3.6.2.3 JULGAMENTO

Essa tarefa visa julgar as subcaracterísticas e características baseada nos resultados da pontuação e nos critérios de julgamento definidos previamente. De acordo com os critérios de seleção e o plano de avaliação, as pontuações devem ser totalizadas a nível de característica.

3.6.3 RELATÓRIO DE AVALIAÇÃO

O resultado final do processo de avaliação será um relatório de avaliação que pode relacionar todas as ferramentas avaliadas, ou então, vários relatórios podem ser escritos, cada

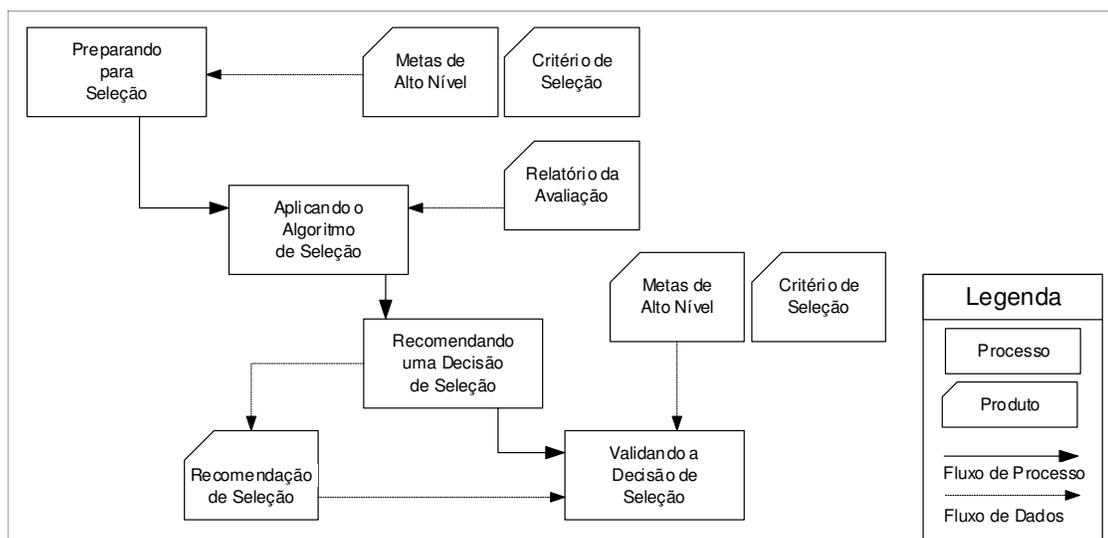
um atuando sobre um subconjunto das ferramentas. O relatório de avaliação deve conter pelo menos as seguintes informações da ferramenta:

- a) nome da ferramenta;
- b) versão;
- c) vendedor;
- d) configuração do ambiente de instalação;
- e) custo dos elementos;
- f) histórico, se apropriado;
- g) fases do ciclo de vida para as quais a ferramenta CASE se aplicará;
- h) modelo de desenvolvimento de software no qual a ferramenta é baseada (ex., modelo em cascata, modelo em espiral);
- i) ambiente de software da ferramenta;
- j) funções da ferramenta;
- k) estrutura de entrada/saída;
- l) público alvo.

O relatório deve discutir as atividades e tarefas específicas no processo de avaliação, em detalhes necessários para permitir o leitor entender o escopo e a profundidade da avaliação e repeti-la, se desejado. Os resultados da avaliação devem ser fornecidos em termos de subcaracterísticas a nível atômico. Baseado nos resultados de menor nível, a agregação poderá ser feita por características de ferramentas CASE segundo consta no item 3.8.

3.7 PROCESSO DE SELEÇÃO

A proposta do processo de seleção é identificar a ferramenta CASE mais adequada entre as candidatas e certificar-se que a ferramenta recomendada atende aos requisitos originais dos usuários. A seleção pode iniciar quando os relatórios de avaliação estiverem concluídos. Um algoritmo de seleção deve ser definido e aplicado aos resultados da avaliação. Uma decisão pode ser recomendada e validada contra o conjunto original de metas e guias de seleção. O processo de seleção é mostrado na figura 5.



Fonte: [ISO98] - Norma ISO/IEC 14102 – Avaliação e seleção de ferramentas CASE

Figura 5 : Visão geral do processo de seleção

3.7.1 PREPARAR PARA A SELEÇÃO

Na preparação para a seleção, devem ser finalizados os critérios de seleção e definido o algoritmo que determina como os dados gerados durante o processo de avaliação são combinados e comparados aos resultados de cada ferramenta candidata avaliada, chegando então a uma decisão.

3.7.2 APLICAR O ALGORITMO DE SELEÇÃO

Os resultados da avaliação são usados como entrada ao algoritmo de seleção, tendo como saída as informações relacionadas às ferramentas candidatas. Cada resultado da avaliação de uma ferramenta fornece um sumário técnico para cada característica da ferramenta, agregado ao nível especificado no algoritmo de seleção (geralmente o nível de característica). O algoritmo de seleção combina o resultado da avaliação das ferramentas candidatas, fornecendo uma comparação para ser usada por quem toma as decisões.

3.7.3 RECOMENDAR UMA DECISÃO DE SELEÇÃO

Quando o algoritmo de seleção for aplicado, uma decisão deve ser feita para adquirir uma ferramenta ou um conjunto delas. Esta é uma decisão gerencial baseada nas comparações técnicas fornecidas pelo processo de avaliação. Deve-se considerar porém, que existem diversos cenários possíveis quando se está avaliando ou selecionando ferramentas CASE. Diferentes metas de negócios devem ser tratadas durante cada um dos processos descritos. A decisão de seleção deve ser justificada com um motivo que resuma a informação e lógica que conduziu para a seleção.

3.7.4 VALIDAR A DECISÃO DE SELEÇÃO

A atividade final no processo deve ser a validação da seleção recomendada. As metas originais da organização, os requisitos dos usuários e os guias de seleção devem ser revisados e comparados aos resultados da avaliação e outros dados relacionando a seleção recomendada. Uma verificação deve ser feita para assegurar que, se a recomendação é aceita, a maioria das metas ou um número suficiente delas serão atingidas.

Pode ser concluído que não existe uma ferramenta adequada. Neste caso, pode ser feita uma escolha entre o desenvolvimento de uma nova ferramenta ou a modificação de uma existente, dentro da própria organização ou fora dela, ou ainda, o abandono de todo o processo de avaliação e seleção.

3.8 CARACTERÍSTICAS DE FERRAMENTAS CASE

As necessidades do usuário que guiam os processos de avaliação e seleção, são baseadas nas características e subcaracterísticas de ferramentas CASE descritas a seguir conforme [ISO98]. Definindo-se as necessidades do usuário, avaliações e comparações poderão ser feitas baseadas em uma visão geral, comum e próxima de um conjunto completo de características. Como foi visto anteriormente, é necessária uma atividade de estruturação para transformar o conjunto de necessidades inicialmente identificadas pelo usuário, em características específicas de ferramentas CASE a fim de que se possa melhor avaliá-las e selecioná-las.

As categorias de avaliação de mais alto nível são chamadas de características. Cada característica é subdividida em subcaracterísticas, que podem ser subdivididas em níveis mais baixos de subcaracterísticas. A seguir, serão definidas as subcaracterísticas atômicas nos termos de seus atributos, onde a cada uma será designado um valor durante o processo de avaliação baseado em uma métrica pré-estabelecida.

Não é comum que algum usuário necessite usar todas as subcaracterísticas atômicas, sendo que os mesmos devem selecionar somente aquelas subcaracterísticas que tem peso significativo com respeito as necessidades de sua organização. Pode haver casos onde necessidades adicionais ou características específicas para uma avaliação ou seleção particular, terão que ser adicionadas. Neste sentido, as subcaracterísticas atômicas devem ser consideradas uma lista parcial, para ser aumentada conforme as necessidades de cada um.

Para subcaracterísticas não-atômicas são fixados valores pela agregação de valores de suas subcaracterísticas atômicas componentes, pesadas conforme definido no plano de avaliação. Esta tarefa de agregação continua até que níveis de agregação do plano de avaliação tenham sido alcançados. O algoritmo de seleção é então usado para combinar os resultados das avaliações de várias ferramentas candidatas para comparação e decisão.

3.8.1 FUNCIONALIDADE – CARACTERÍSTICAS RELACIONADAS AO PROCESSO DE CICLO DE VIDA

As características de ferramentas CASE integrantes nesse conjunto tratam da funcionalidade relacionada ao ciclo de vida da Engenharia de Software. Para os processos de ciclo de vida referenciados, as definições da [NBR97] se aplicam. A tabela 6 apresenta a relação dessas características e suas subcaracterísticas atômicas.

Tabela 6 : Características relacionadas ao processo de ciclo de vida

Característica: Processo de gerenciamento
<u>Subcaracterísticas atômicas:</u>
a) estimativa de custo e programação;
b) planejamento;
c) trajetória do projeto;

<p>d) análise do estado do projeto e relatório;</p> <p>e) gerência de processos.</p>
<p>Característica: Processo de desenvolvimento – modelagem</p>
<p><u>Subcaracterísticas atômicas:</u></p>
<p>a) desenvolvimento de diagrama;</p> <p>b) análise de diagrama;</p> <p>c) suporte na especificação de requisitos;</p> <p>d) suporte em especificação de desenho;</p> <p>e) modelagem de especificação de construção;</p> <p>f) simulação;</p> <p>g) prototipação;</p> <p>h) modelagem de interface.</p>
<p>Característica: Processo de desenvolvimento – construção</p>
<p><u>Subcaracterísticas atômicas:</u></p>
<p>a) geração de código;</p> <p>b) geração de esquemas de banco de dados;</p> <p>c) geração de telas;</p> <p>d) geração de relatórios;</p> <p>e) compilação;</p> <p>f) sintaxe do editor;</p> <p>g) <i>debugging</i>.</p>
<p>Característica: Processo de manutenção</p>
<p><u>Subcaracterísticas atômicas:</u></p>
<p>a) entendimento do problema;</p> <p>b) localização;</p> <p>c) análise de impacto;</p> <p>d) engenharia de dados reversa;</p> <p>e) processo e procedimento de engenharia reversa;</p> <p>f) reestruturação de código fonte;</p> <p>g) tradução de código fonte.</p>

Característica: Processo de documentação
<u>Subcaracterísticas atômicas:</u>
<ul style="list-style-type: none"> a) edição de texto; b) edição gráfica; c) edição baseada em formatos; d) editoração; e) suporte para hipertexto; f) manipulação de mudanças; g) extração automática de dados e geração de documentação.
Característica: Processo de gerenciamento de configuração
<u>Subcaracterísticas atômicas:</u>
<ul style="list-style-type: none"> a) controle de acesso; b) registro de modificações; c) definições e gerenciamento de múltiplas versões; d) contabilidade das informações de configuração; e) geração de <i>releases</i>; f) capacidade de armazenamento.
Característica: Processo de certificação de qualidade
<u>Subcaracterísticas atômicas:</u>
<ul style="list-style-type: none"> a) gerenciamento da qualidade dos dados; b) gerenciamento de riscos.
Característica: Processo de verificação
<u>Subcaracterísticas atômicas:</u>
<ul style="list-style-type: none"> a) análise de especificações de rastreamento; b) análise de especificação; c) análise de código fonte.
Característica: Processo de validação
<u>Subcaracterísticas atômicas:</u>
<ul style="list-style-type: none"> a) técnicas de prova de exatidão; b) análise de falhas; c) análise de defeitos;

- d) casos de teste e entrada de resultados esperados;
- e) casos de teste e geração de resultados esperados;
- f) rastreabilidade de teste;
- g) instrumentação de código fonte;
- h) captura e reapresentação de entradas;
- i) testes dirigidos;
- j) análise de execução;
- k) análise de confiabilidade;
- l) análise de testes de cobertura;
- m) administração de procedimentos de testes;
- n) testes de regressão;
- o) verificação automática de resultados;
- p) análise estatística de testes;
- q) simulação de ambiente de operação;
- r) integração de testes.

Fonte: Características de ferramentas CASE – adaptada de [ISO98]

3.8.2 FUNCIONALIDADE – CARACTERÍSTICAS RELATADAS NO USO DA FERRAMENTA CASE

As características de ferramentas CASE integrantes nesse conjunto, tratam da funcionalidade relacionada ao ambiente no qual a ferramenta opera, à integrabilidade de ferramentas CASE e aos aspectos da aplicação de ferramentas CASE. Essas características relacionam a ferramenta e seu ambiente e os projetos nos quais essas serão suportadas. A tabela 7 apresenta a relação dessas características e suas subcaracterísticas atômicas.

Tabela 7 : Características relacionadas no uso da ferramenta CASE

Característica: Ambiente no qual a ferramenta CASE opera
<u>Subcaracterísticas atômicas:</u>
<ul style="list-style-type: none"> a) características de hardware requerido pela ferramenta; b) ambiente de software requerido pela ferramenta; c) repositório de software (base de informações); d) ambiente físico da ferramenta.

Característica: Integrabilidade da ferramenta CASE
<u>Subcaracterísticas atômicas:</u>
<ul style="list-style-type: none"> a) compatibilidade com elementos do ambiente; b) integração de dados; c) integração de controle; d) integração de apresentação; e) acesso a metadados.
Característica: Aspectos da aplicação da ferramenta CASE
<u>Subcaracterísticas atômicas:</u>
<ul style="list-style-type: none"> a) ambiente de hardware e software dos produtos da ferramenta; b) conformidade com normas; c) domínio da aplicação; d) linguagens suportadas; e) bancos de dados suportados; f) suporte metodológico.

Fonte: Características de ferramentas CASE – adaptada de [ISO98]

3.8.3 CARACTERÍSTICAS GERAIS DE QUALIDADE

As características de ferramentas CASE integrantes nesse conjunto, tratam da verificação de atributos relacionados à qualidade dessas ferramentas. Para a avaliação de características de qualidade, as definições contidas na [NBR96] e em [WEB94] se aplicam. A tabela 8 apresenta a relação dessas características e suas subcaracterísticas atômicas.

Tabela 8 : Características gerais de qualidade

Característica: Funcionalidade (satisfaz as necessidades ?)
<u>Subcaracterísticas atômicas:</u>
<ul style="list-style-type: none"> a) adequação; b) acurácia; c) interoperabilidade; d) conformidade; e) segurança de acesso.

Característica: Confiabilidade (é imune a falhas ?)
<u>Subcaracterísticas atômicas:</u>
a) maturidade; b) tolerância a falhas; c) recuperabilidade.
Característica: Usabilidade (é fácil de usar ?)
<u>Subcaracterísticas atômicas:</u>
a) intelegibilidade; b) apreensibilidade; c) operacionalidade.
Característica: Eficiência (é rápida e “enxuta” ?)
<u>Subcaracterísticas atômicas:</u>
a) tempo; b) recursos.
Característica: Manutenibilidade (é fácil de modificar ?)
<u>Subcaracterísticas atômicas:</u>
a) analisabilidade; b) modificabilidade; c) estabilidade; d) testabilidade.
Característica: Portabilidade (é fácil de usar em outro ambiente ?)
<u>Subcaracterísticas atômicas:</u>
a) adaptabilidade; b) capacidade para ser instalada; c) conformidade; d) capacidade de substituir.

Fonte: Características de ferramentas CASE – adaptada de [ISO98]

4 DESCRIÇÃO DO SOFTWARE

4.1 INTRODUÇÃO

Foi desenvolvido um software para apoiar e facilitar a tarefa de avaliação e seleção de ferramentas CASE. O software de acordo com a norma ISO/IEC 14102, apoia algumas tarefas relacionadas com os seus quatro processos que são:

- a) processo de iniciação: apoiar na elaboração das metas da organização, bem como no armazenamento dos dados referentes às ferramentas candidatas, seus fornecedores e avaliadores;
- b) processo de estruturação: apoiar na elaboração dos requisitos da organização e na montagem e emissão do questionário de avaliação com a atribuição dos respectivos pesos de cada subcaracterística atômica de ferramenta avaliada;
- c) processo de avaliação: apoiar o registro das notas atribuídas por cada avaliador a cada subcaracterística atômica de ferramenta avaliada, listar esse questionário já com suas notas para ser apresentado à organização para conferência e correção caso existam erros;
- d) processo de seleção: executar o algoritmo de seleção que tem como entrada as notas atribuídas às subcaracterísticas atômicas, multiplica-as pelos seus respectivos pesos atribuídos e acumula a pontuação em cada ferramenta. Após a execução do algoritmo de seleção, pode-se emitir uma listagem das ferramentas avaliadas com seus dados básicos e sua pontuação acumulada para que se processe a seleção da ferramenta que melhor atende os requisitos da organização.

O software foi especificado seguindo a Análise Essencial de Sistemas, sendo para isso utilizada a ferramenta CASE *Power Designer 6.1* da *Sybase Inc.*, versão demonstração. Fazem parte da especificação do software, a descrição do software e do seu objetivo, a lista de eventos, o Diagrama de Contexto, o Diagrama de Fluxo de Dados nos níveis 1 e 2, o Modelo de Entidades-Relacionamentos e o Dicionário de Dados que apresenta as entidades com seus atributos. A implementação foi construída no ambiente de programação visual *Delphi 4* da *Borland International Inc.*

O software tem por objetivo apoiar a avaliação e seleção de ferramentas CASE, através da utilização de questionários.

4.2 LISTA DE EVENTOS

O software possui doze eventos que são apresentados a seguir:

1. Fornecedor é cadastrado;
2. Fornecedor indica ferramenta;
3. Avaliador é cadastrado;
4. Organização define características de ferramentas CASE;
5. Organização define subcaracterísticas de ferramentas CASE;
6. Organização define metas;
7. Organização define requisitos;
8. Organização define tabelas;
9. Organização recebe questionário de avaliação;
10. Avaliador informa resultado da avaliação;
11. Organização recebe relação da avaliação;
12. Organização recebe relação de ferramentas CASE com pontuação.

4.3 DIAGRAMA DE CONTEXTO

O Diagrama de Contexto estabelece os limites entre o sistema e o seu ambiente. É utilizado para mostrar as comunicações entre o sistema, o ambiente e as entidades com as quais se comunica. A figura 6 apresenta o Diagrama de Contexto do software.

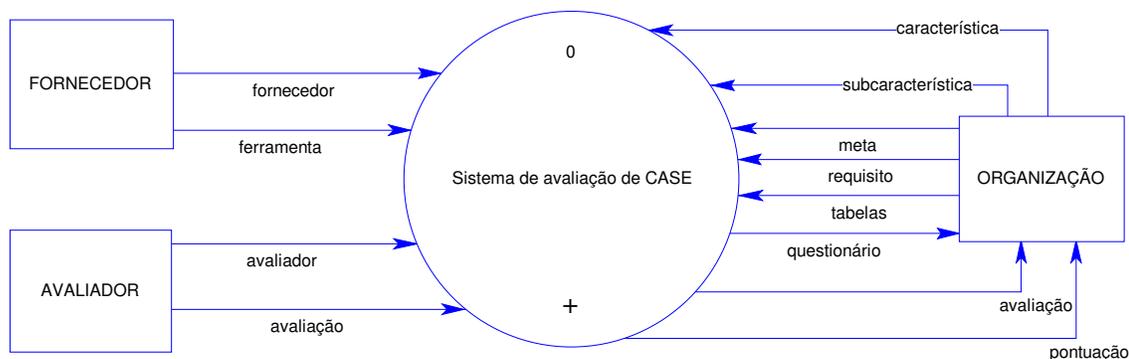


Figura 6 : Diagrama de Contexto

4.4 DIAGRAMA DE FLUXO DE DADOS

O objetivo do Diagrama de Fluxo de Dados é mostrar um sistema completo ou parte dele, de onde os dados surgem, para onde vão, quando são armazenados, que processos os transformam e as iterações entre armazenamento de dados e processos. O DFD permite a avaliação do modelo junto ao usuário, a fim de identificar falhas o mais cedo possível no projeto para que as mesmas sejam corrigidas. As figuras 7 e 8 apresentam o Diagrama de Fluxo de Dados do software, nível 1 e nível 2.

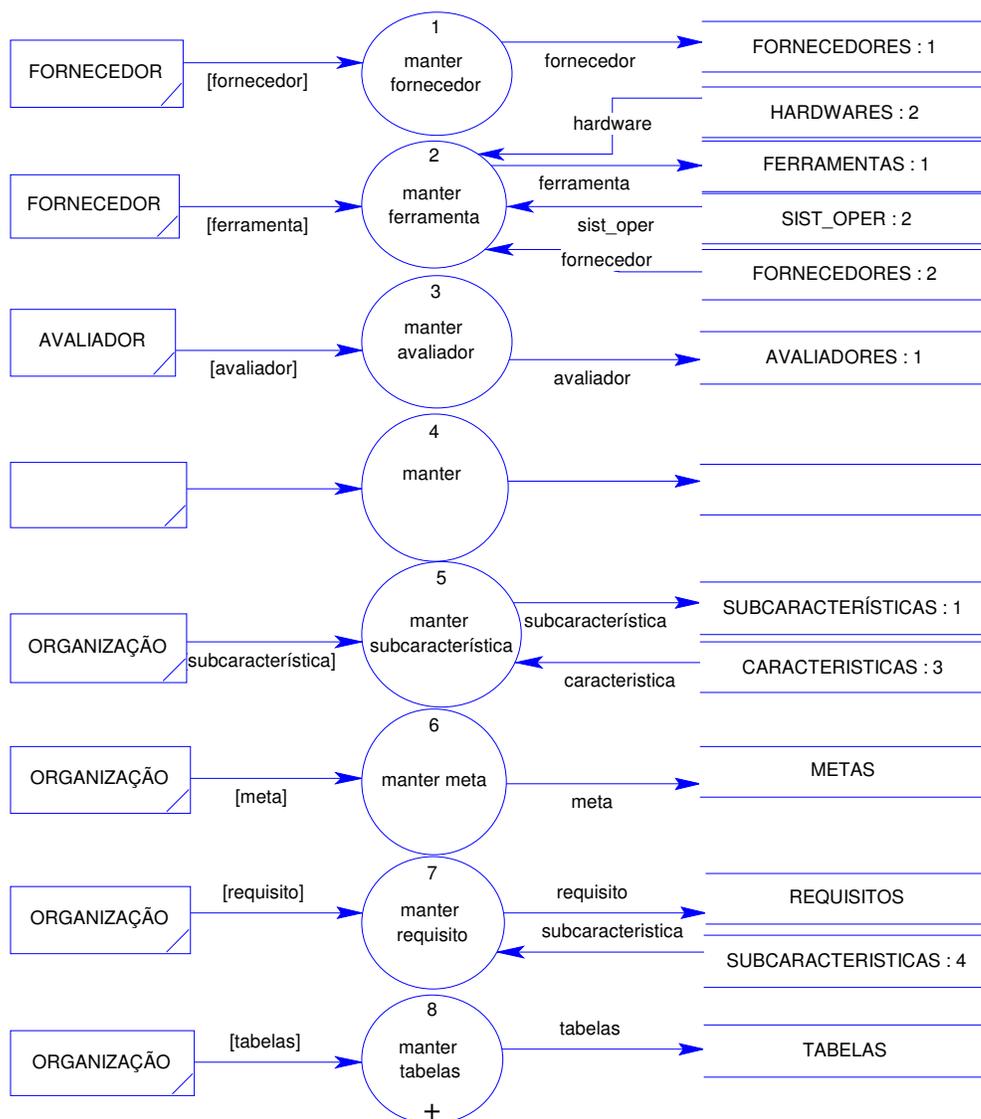


Figura 7 : Diagrama de Fluxo de Dados – nível 1

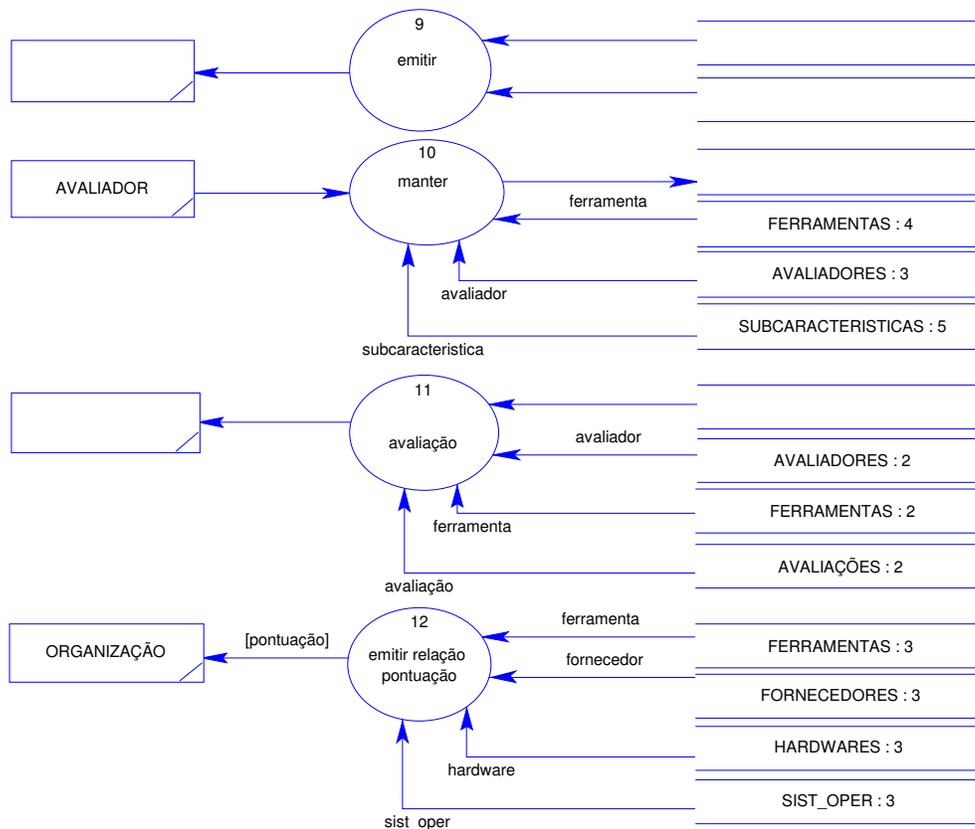


Figura 7 : Diagrama de Fluxo de Dados – nível 1 - continuação

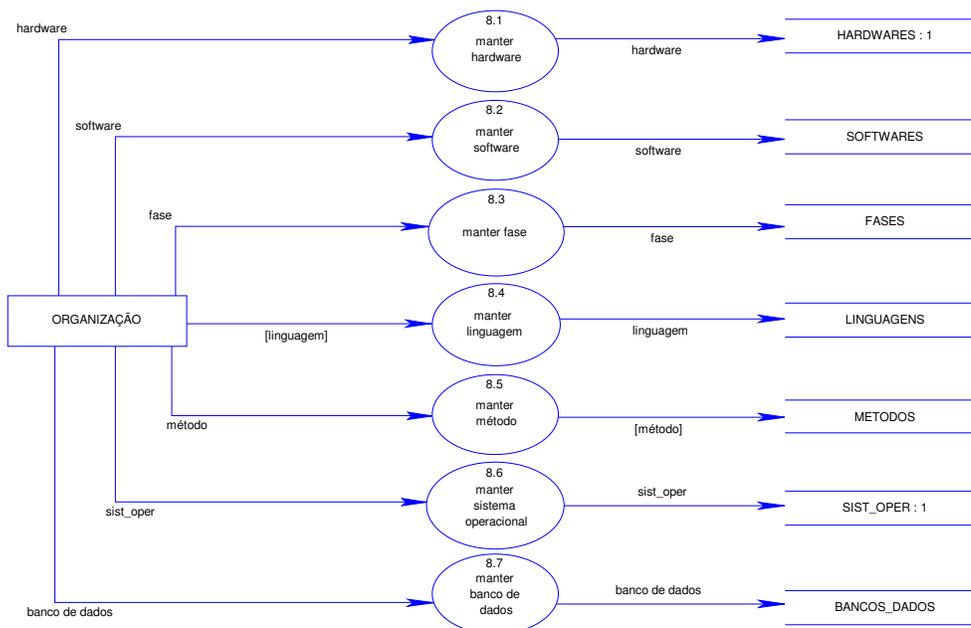


Figura 8 : Diagrama de Fluxo de Dados – nível 2

4.5 MODELO ENTIDADE-RELACIONAMENTO

O Modelo Entidade-Relacionamento é um diagrama utilizado para detalhar as associações existentes entre as entidades de dados do sistema. A figura 9 apresenta o Modelo Entidade-Relacionamento do software.

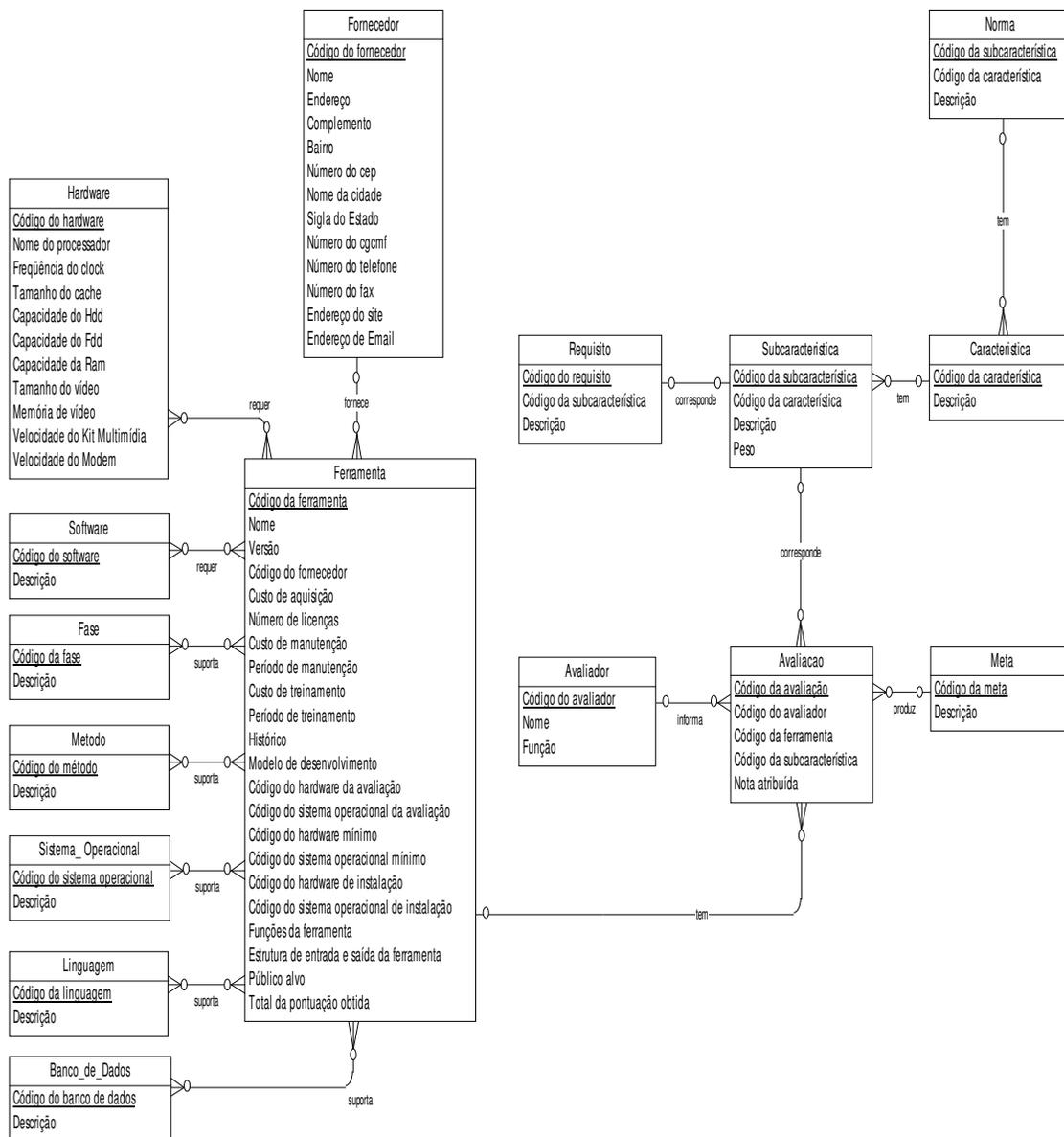


Figura 9 : Modelo Entidade-Relacionamento

4.6 DICIONÁRIO DE DADOS

O Dicionário de Dados fornece a informação de texto de suporte para complementar a informação gráfica mostrada no Diagrama de Fluxo de Dados – DFD, sendo considerado um grupo organizado de definições, de todos os elementos de dados do sistema sendo modelado.

Para a documentação do Dicionário de Dados é utilizado o seguinte formato:

- a) nome e descrição da tabela;
- b) a coluna *name* apresenta uma breve descrição do atributo;
- c) a coluna *code* apresenta o nome que identifica o atributo na tabela;
- d) a coluna *type* apresenta o tipo do atributo, que pode ser: I – número inteiro, A – alfanumérico, MN – moeda e N – numérico com possibilidade de atribuição de casas decimais;
- e) a coluna *I* identifica se o atributo é chave primária da tabela;
- f) a coluna *M* identifica se é obrigatório o preenchimento do atributo.

São listadas a seguir as tabelas utilizadas pelo software desenvolvido, apresentando seus dados conforme descrição acima.

FERRAMENTA: Tabela contendo os atributos de ferramentas CASE

Tabela 9 : Tabela de ferramentas

Name	Code	Type	I	M
Código da ferramenta	CODIGO_FERRAMENTA	I	Yes	Yes
Nome	NOME	A30	No	No
Versão	VERSAO	A10	No	No
Código do fornecedor	COD_FORNECEDOR	I	No	No
Custo de aquisição	CUSTO_AQUIS	MN6,2	No	No
Número de licenças	NUM_LICENCAS	I	No	No
Custo de manutenção	CUSTO_MANUT	MN6,2	No	No
Período de manutenção	PER_MANUT	A10	No	No
Custo de treinamento	CUSTO_TREINAM	MN6,2	No	No

Name	Code	Type	I	M
Período de treinamento	PER_TREINAM	A10	No	No
Histórico	HISTORICO	A30	No	No
Modelo de desenvolvimento	MOD_DESENV	A10	No	No
Código do hardware da avaliação	COD_HARD_AVAL	I	No	No
Código do sistema operacional da avaliação	COD_SIST_OPER_AVAL	I	No	No
Código do hardware mínimo	COD_HARD_MINIMO	I	No	No
Código do sistema operacional mínimo	COD_SIST_OPER_MIN	I	No	No
Código do hardware de instalação	COD_HARD_INST	I	No	No
Código do sistema operacional de instalação	COD_SIST_OPER_INST	I	No	No
Funções da ferramenta	FUNCOES	A50	No	No
Estrutura de entrada e saída da ferramenta	ESTRUTURA_ES	A50	No	No
Público alvo	PUBL_ALVO	A50	No	No
Total da pontuação obtida	PONTUACAO	I	No	No

FORNECEDOR: Tabela contendo os atributos de fornecedores de ferramentas
CASE

Tabela 10 : Tabela de fornecedores

Name	Code	Type	I	M
Código do fornecedor	COD_FORN	I	Yes	Yes
Nome	NOME	A30	No	No
Endereço	ENDERECO	A20	No	No
Complemento	COMPLEMENTO	A15	No	No
Bairro	BAIRRO	A20	No	No
Número do cep	CEP	A10	No	No
Nome da cidade	CIDADE	A30	No	No
Sigla do Estado	ESTADO	A2	No	No
Número do cgcmf	CGCMF	A18	No	No
Número do telefone	TELEFONE	A14	No	No

Name	Code	Type	I	M
Número do fax	FAX	A14	No	No
Endereço do site	WEBSITE	A40	No	No
Endereço de Email	EMAIL	A40	No	No

AVALIADOR: Tabela contendo os atributos dos avaliadores de ferramentas CASE

Tabela 11 : Tabela de avaliadores

Name	Code	Type	I	M
Código do avaliador	COD_AVAL	I	Yes	Yes
Nome	NOME	A30	No	No
Função	FUNCAO	A25	No	No

CARACTERISTICA: Tabela contendo os atributos de características de ferramentas CASE

Tabela 12 : Tabela de características

Name	Code	Type	I	M
Código da característica	COD_CAR	I	Yes	Yes
Descrição	DESCRICA0	A40	No	No

SUBCARACTERISTICA: Tabela contendo os atributos de subcaracterísticas de ferramentas CASE

Tabela 13 : Tabela de subcaracterísticas

Name	Code	Type	I	M
Código da subcaracterística	COD_SUBC	I	Yes	Yes
Código da característica	COD_CARSC	I	No	No
Descrição	DESCRICA0	A40	No	No
Peso	PESO	I	No	No

AVALIACAO: Tabela contendo os atributos de resultado de avaliações de ferramentas CASE

Tabela 14 : Tabela de avaliações

Name	Code	Type	I	M
Código da avaliação	COD_AVALIACAO	I	Yes	Yes
Código do avaliador	COD_AVALIADOR	I	No	No
Código da ferramenta	COD_FERR	I	No	No
Código da subcaracterística	COD_SUBCAR	I	No	No
Nota atribuída	NOTA	I	No	No

HARDWARE: Tabela contendo os atributos de configurações de hardware

Tabela 15 : Tabela de configurações de hardware

Name	Code	Type	I	M
Código do hardware	COD_HARD	I	Yes	Yes
Nome do processador	PROCESSADOR	A15	No	No
Frequência do clock	CLOCK	I	No	No
Tamanho do cache	CACHE	I	No	No
Capacidade do HDD	HDD	N5,2	No	No
Capacidade do FDD	FDD	N5,2	No	No
Capacidade da Ram	RAM	I	No	No
Tamanho do vídeo	VIDEO	I	No	No
Memória do vídeo	MEMVIDEO	I	No	No
Velocidade do Kit Multimídia	MULTIMIDIA	I	No	No
Velocidade do Modem	MODEM	N5,2	No	No

REQUISITO: Tabela contendo os atributos de requisitos da organização

Tabela 16 : Tabela de requisitos

Name	Code	Type	I	M
Código do requisito	COD_REQ	I	Yes	Yes
Código da subcaracterística	COD_SUBCAR	I	No	No
Descrição	DESCRICAO	A40	No	No

META: Tabela contendo os atributos de metas da organização

Tabela 17 : Tabela de metas

Name	Code	Type	I	M
Código da meta	COD_META	I	Yes	Yes
Descrição	DESCRICA0	A40	No	No

NORMA: Tabela contendo os atributos de subcaracterísticas de ferramentas CASE apresentados pela norma ISO/IEC 14102

Tabela 18 : Tabela de subcaracterísticas da norma ISO/IEC 14102

Name	Code	Type	I	M
Código da subcaracterística	COD_SUBCN	I	Yes	Yes
Código da característica	COD_CARN	I	No	No
Descrição	DESCRICA0	A40	No	No

FASE: Tabela contendo os atributos de fases do ciclo de vida da Engenharia de Software

Tabela 19 : Tabela de fases do ciclo de vida da Engenharia de Software

Name	Code	Type	I	M
Código da fase	COD_FASE	I	Yes	Yes
Descrição	DESCRICA0	A40	No	No

LINGUAGEM: Tabela contendo os atributos de linguagens de programação

Tabela 20 : Tabela de linguagens de programação

Name	Code	Type	I	M
Código da linguagem	COD_LING	I	Yes	Yes
Descrição	DESCRICA0	A40	No	No

METODO: Tabela contendo os atributos de metodologias de desenvolvimento de software

Tabela 21 : Tabela de metodologias de desenvolvimento de software

Name	Code	Type	I	M
Código do método	COD_MET	I	Yes	Yes
Descrição	DESCRICA0	A40	No	No

SISTEMAOPERACIONAL: Tabela contendo os atributos de sistemas operacionais

Tabela 22 : Tabela de sistemas operacionais

Name	Code	Type	I	M
Código do sistema operacional	COD_SISOP	I	Yes	Yes
Descrição	DESCRICA0	A40	No	No

BANCODADOS: Tabela contendo os atributos de bancos de dados

Tabela 23 : Tabela de bancos de dados

Name	Code	Type	I	M
Código do banco de dados	COD_BD	I	Yes	Yes
Descrição	DESCRICA0	A40	No	No

SOFTWARE: Tabela contendo os atributos de software

Tabela 24 : Tabela de softwares

Name	Code	Type	I	M
Código do software	COD_SOFT	I	Yes	Yes
Descrição	DESCRICA0	A40	No	No

4.7 PRINCIPAIS TELAS, RELATÓRIOS E GRÁFICO DO SOFTWARE

O software desenvolvido apresenta diversas telas, alguns relatórios e um gráfico. Entre as telas se pode citar a de manutenção da tabela de ferramentas, a de montagem do questionário de avaliação e a do registro das notas atribuídas às subcaracterísticas. Entre os relatórios tem-se o questionário de avaliação e por fim, apresenta-se o gráfico do nível de

atendimento dos requisitos da organização. A seguir são apresentadas as principais telas, relatórios e o gráfico do software. A figura 10 apresenta a tela inicial do software.



Figura 10 : Tela inicial do software

A figura 11 apresenta a tela de manutenção de ferramentas CASE.

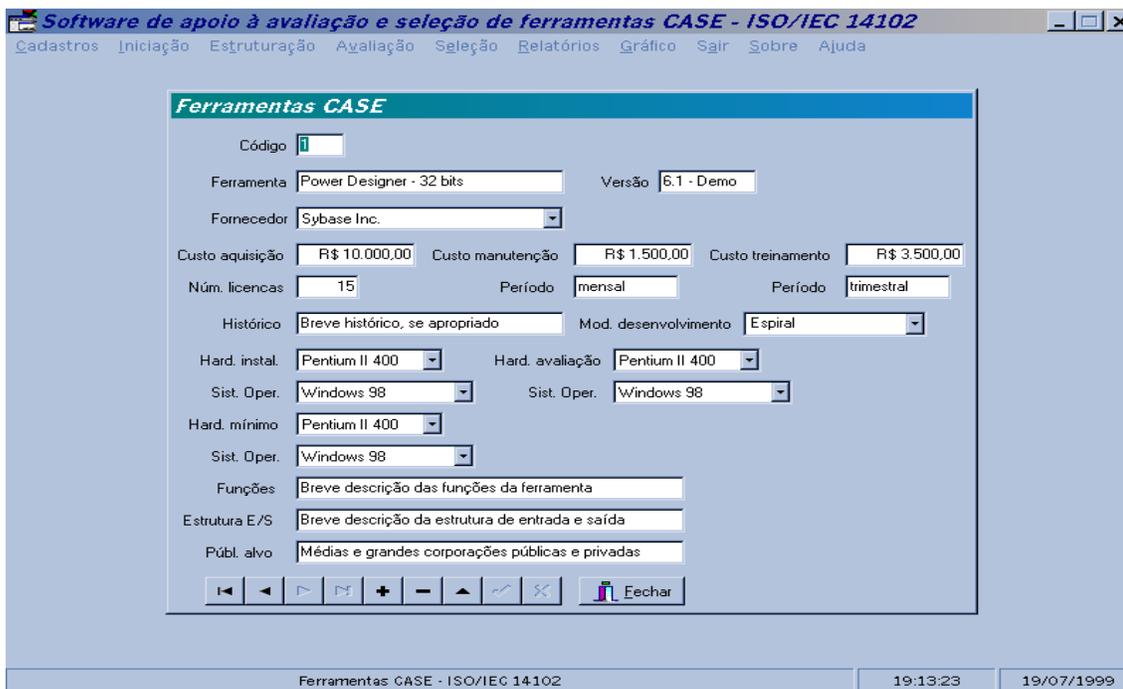


Figura 11 : Manutenção de ferramentas CASE

A figura 12 apresenta a tela de manutenção das subcaracterísticas de ferramentas CASE que serão avaliadas, bem como a atribuição de pesos a cada uma.

Questionário de avaliação de ferramentas CASE

Cód. subc.	Cód. car.	Característica	Subcaracterística	Peso
1	1	Proc. de gerenciamento	Estimativa de custo e programação	1
2	1	Proc. de gerenciamento	Planejamento	1
3	1	Proc. de gerenciamento	Trajectoria do projeto	1
4	1	Proc. de gerenciamento	Análise do status do projeto e relatório	1
5	1	Proc. de gerenciamento	Gerência de processos	1
6	2	Proc. de desenvolvimento - modelagem	Desenvolvimento de diagrama	1
7	2	Proc. de desenvolvimento - modelagem	Análise de diagrama	1
8	2	Proc. de desenvolvimento - modelagem	Suporte na especificação de requisitos	1
9	2	Proc. de desenvolvimento - modelagem	Suporte na especificação de desenho	1
10	2	Proc. de desenvolvimento - modelagem	Modelagem de especificação de construção	1
11	2	Proc. de desenvolvimento - modelagem	Simulação	1
12	2	Proc. de desenvolvimento - modelagem	Prototipação	1
13	2	Proc. de desenvolvimento - modelagem	Modelagem de interface	1
14	3	Proc. de desenvolvimento - construção	Geração de código	1
15	3	Proc. de desenvolvimento - construção	Geração de esquemas de banco de dados	1
16	3	Proc. de desenvolvimento - construção	Geração de telas	1
17	3	Proc. de desenvolvimento - construção	Geração de relatórios	1
18	3	Proc. de desenvolvimento - construção	Compilação	1
19	3	Proc. de desenvolvimento - construção	Sintaxe do editor	1
20	3	Proc. de desenvolvimento - construção	Debugging	1
21	4	Manutenção de processos	Entendimento do problema	1
22	4	Manutenção de processos	Localização	1
23	4	Manutenção de processos	Análise de impacto	1
24	4	Manutenção de processos	Engenharia de dados reversa	1
25	4	Manutenção de processos	Proc. e procedim. de engenharia reversa	1
26	4	Manutenção de processos	Reestruturação do código fonte	1

Figura 12 : Manutenção das subcaracterísticas de ferramentas CASE

A figura 13 apresenta a tela onde são registradas as notas atribuídas a cada subcaracterística atômica de ferramenta CASE, dada por cada avaliador.

Resultado da avaliação

Código	Ferramenta	Avaliador	Subcaracterística	Nota
1	Power Designer - 32 bits	Jair Weinrich	Estimativa de custo e programação	5
2	Power Designer - 32 bits	Jair Weinrich	Planejamento	5
3	Power Designer - 32 bits	Jair Weinrich	Trajectoria do projeto	5
4	Power Designer - 32 bits	Jair Weinrich	Análise do status do projeto e relatório	5
5	Power Designer - 32 bits	Jair Weinrich	Gerência de processos	7
6	Power Designer - 32 bits	Jair Weinrich	Desenvolvimento de diagrama	9
7	Power Designer - 32 bits	Jair Weinrich	Análise de diagrama	9
8	Power Designer - 32 bits	Jair Weinrich	Suporte na especificação de requisitos	5
9	Power Designer - 32 bits	Jair Weinrich	Suporte na especificação de desenho	5
10	Power Designer - 32 bits	Jair Weinrich	Modelagem de especificação de construção	7
11	Power Designer - 32 bits	Jair Weinrich	Simulação	5
12	Power Designer - 32 bits	Jair Weinrich	Prototipação	5
13	Power Designer - 32 bits	Jair Weinrich	Modelagem de interface	5
14	Power Designer - 32 bits	Jair Weinrich	Geração de código	5
15	Power Designer - 32 bits	Jair Weinrich	Geração de esquemas de banco de dados	5
16	Power Designer - 32 bits	Jair Weinrich	Geração de telas	5
17	Power Designer - 32 bits	Jair Weinrich	Geração de relatórios	5
18	Power Designer - 32 bits	Jair Weinrich	Compilação	5
19	Power Designer - 32 bits	Jair Weinrich	Sintaxe do editor	5
20	Power Designer - 32 bits	Jair Weinrich	Debugging	5
21	Power Designer - 32 bits	Jair Weinrich	Entendimento do problema	5
22	Power Designer - 32 bits	Jair Weinrich	Localização	5
23	Power Designer - 32 bits	Jair Weinrich	Análise de impacto	5
24	Power Designer - 32 bits	Jair Weinrich	Engenharia de dados reversa	6
25	Power Designer - 32 bits	Jair Weinrich	Proc. e procedim. de engenharia reversa	6
26	Power Designer - 32 bits	Jair Weinrich	Reestruturação do código fonte	5

Figura 13 : Manutenção das notas atribuídas às subcaracterísticas de ferramentas CASE

A figura 14 apresenta o questionário de avaliação das ferramentas.

19:14
19/07/1999

QUESTIONÁRIO DE AVALIAÇÃO DE FERRAMENTAS CASE

Característica: 1 - Proc. de gerenciamento

Subcaracterística	Peso (de 1 a 5)	Nota (de 1 a 10)
1 - Estimativa de custo e programação	1	_____
2 - Planejamento	1	_____
3 - Trajetória do projeto	1	_____
4 - Análise do status do projeto e relatório	1	_____
5 - Gerência de processos	1	_____

Característica: 2 - Proc. de desenvolvimento - modelagem

Subcaracterística	Peso (de 1 a 5)	Nota (de 1 a 10)
6 - Desenvolvimento de diagrama	1	_____
7 - Análise de diagrama	1	_____
8 - Suporte na especificação de requisitos	1	_____
9 - Suporte na especificação de desenho	1	_____
10 - Modelagem de especificação de construção	1	_____

0% Page 1 of 5

Ferramentas CASE - ISO/IEC 14102

19:15:04 19/07/1999

Figura 14 : Questionário de avaliação das ferramentas

A figura 15 apresenta a relação das subcaracterísticas atômicas de ferramentas CASE com as respectivas notas atribuídas.

19:15
19/07/1999

RELAÇÃO DA AVALIAÇÃO DE FERRAMENTAS CASE

Ferramenta:	1 - Power Designer -32 bits	Avaliador:	1 - Jair Weinrich
Resultado	Subcaracterística		Nota atribuída
1	1 - Estimativa de custo e programação		5
2	2 - Planejamento		5
3	3 - Trajetória do projeto		5
4	4 - Análise do status do projeto e relatório		5
5	5 - Gerência de processos		7
6	6 - Desenvolvimento de diagrama		9
7	7 - Análise de diagrama		9
8	8 - Suporte na especificação de requisitos		5
9	9 - Suporte na especificação de desenho		5
10	10 - Modelagem de especificação de construção		7
11	11 - Simulação		5
12	12 - Prototipação		5
13	13 - Modelagem de interface		5

0% Page 1 of 5

Ferramentas CASE - ISO/IEC 14102

19:15:42 19/07/1999

Figura 15 : Relação das subcaracterísticas com as notas atribuídas

A figura 16 apresenta a relação das ferramentas CASE com seus dados básicos e sua respectiva totalização da pontuação obtida na avaliação.

The screenshot shows a 'Print Preview' window with a title bar containing 'Print Preview' and standard window controls. The main content area displays a report titled 'RELAÇÃO DE FERRAMENTAS CASE' with a timestamp of 19:15 on 19/07/1999. The report details the following information:

Ferramenta: 1 - Power Designer - 32 bits			
Versão.....	6.1 - Demo	Fornecedor.....	1 - Sybase Inc.
Custo de aquisição.....	R\$ 10.000,00	Número de licenças:	15
Custo de manutenção.....	R\$ 1.500,00	Periodicidade.....	mensal
Custo de treinamento.....	R\$ 3.500,00	Período.....	trimestral
Hardware de instalação.....	Pentium II 400	Sistema operacional:	Windows 98
Hardware da avaliação.....	Pentium II 400	Sistema operacional:	Windows 98
Hardware mínimo.....	Pentium II 400	Sistema operacional:	Windows 98
Histórico.....	Breve histórico, se apropriado		
Modelo de desenvolvimento:	Espiral		
Funções.....	Breve descrição das funções da ferramenta		
Estrutura de E/S.....	Breve descrição da estrutura de entrada e saída		
Público alvo.....	Médias e grandes corporações públicas e privadas		
Resultado da avaliação.....	604		

At the bottom of the window, the status bar shows '0%' zoom, 'Page 1 of 1', 'Ferramentas CASE - ISO/IEC 14102', '19:16:11', and '19/07/1999'.

Figura 16 : Relação de ferramentas CASE com a pontuação obtida

A organização, antes de avaliar ferramentas CASE para selecionar aquela que melhor atende às suas necessidades de desenvolvimento e manutenção de software, deverá definir requisitos a serem atendidos que irão nortear os processos de avaliação e seleção. Cada um desses requisitos deverá fazer referência a uma subcaracterística de ferramenta CASE. Durante o processo de avaliação, todas as subcaracterísticas receberão uma nota que será atribuída por cada avaliador de ferramentas CASE. A partir das notas atribuídas a essas subcaracterísticas, considerando a ferramenta que obteve a maior pontuação dentre as candidatas avaliadas, define-se o nível de atendimento dos requisitos da organização que irá auxiliar na decisão a ser tomada pela organização. A organização poderá definir valores mínimos aceitáveis para esse nível, bem como definir uma média aritmética mínima aceitável que será obtida através da soma das notas atribuídas às subcaracterísticas de CASE relacionadas com os requisitos da organização e dividida pelo número total dessas ocorrências de notas. De acordo com esse nível, a organização, ao compará-lo com os parâmetros estabelecidos para o mesmo, deverá decidir: pela aquisição da ferramenta selecionada, por novas avaliações ou tomar outra atitude que julgar mais conveniente. A figura 17 apresenta o gráfico do nível de atendimento dos requisitos da organização.

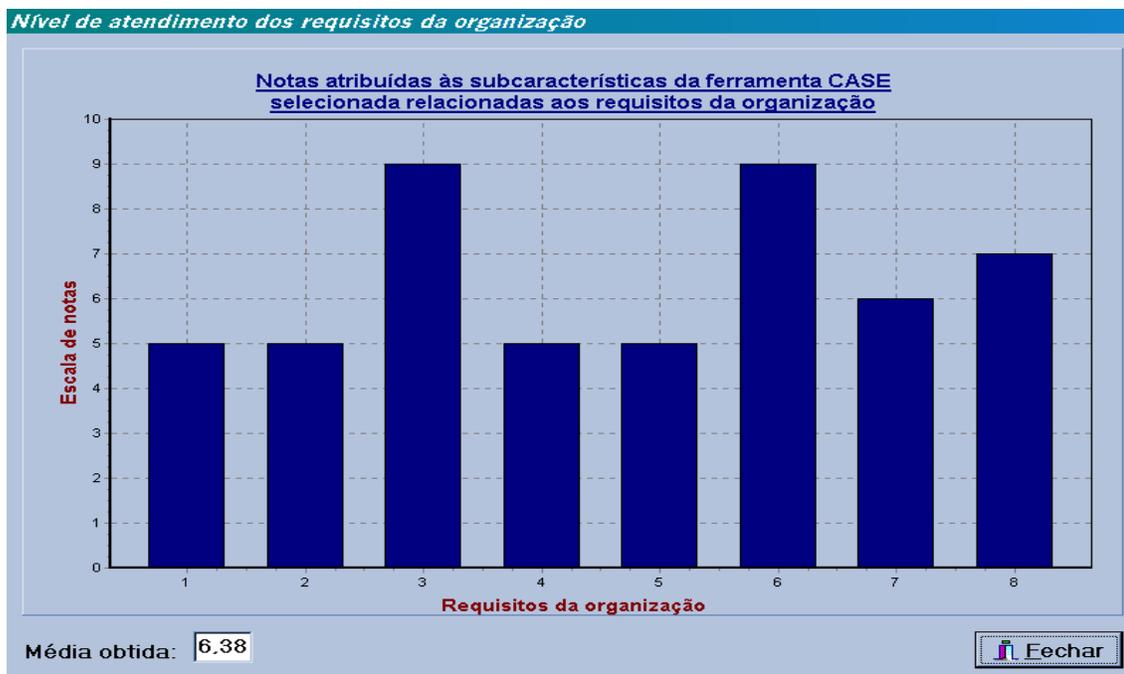


Figura 17 : Nível de atendimento dos requisitos da organização

4.8 APLICAÇÃO DO SOFTWARE

Este software foi disponibilizado para ser adotado na disciplina de Engenharia de Software da FURB, durante o primeiro semestre de 1999, para avaliação de algumas ferramentas CASE: *Designer 2000*, *Dr. CASE*, *ErWin*, *Genexus*, *Playground*, *Power Designer*, *Rational Rose* e *System Architect*. O objetivo dessa avaliação foi aproximar os alunos de ferramentas de desenvolvimento de software, para que obtivessem os subsídios necessários para o uso das mesmas. Em nenhum momento pensou-se em comparar essas ferramentas, apenas buscou-se uma familiarização com as mesmas, com o intuito de conhecê-las melhor ou mesmo manter um primeiro contato, visto que grande percentual dos alunos envolvidos, possuía pouco ou nenhum conhecimento acerca das ferramentas avaliadas. A partir do seu uso, o software demonstrou sua aplicabilidade ao facilitar a adoção da norma ISO/IEC 14102 – Avaliação e seleção de ferramentas CASE por parte dos alunos que a utilizaram didaticamente, inclusive sugerindo melhorias a serem implementadas. O software também foi utilizado pelo autor do presente trabalho, na avaliação também didática da ferramenta CASE *Power Designer 6.1* versão demonstração, da *Sybase Inc*.

5 CONCLUSÕES

5.1 CONSIDERAÇÕES FINAIS

A utilização de metodologias de desenvolvimento de software constitui fator fundamental para a obtenção de sucesso durante o ciclo de vida do software. Uma organização que desenvolve software sem a utilização dessas metodologias, encontra dificuldades devido ao fato de que depende unicamente das habilidades e competências de sua equipe de desenvolvimento. Metodologias propõem a adoção de padrões e métodos, provendo parâmetros gerais e específicos a seguir na tarefa de desenvolver software, onde toda a equipe de desenvolvimento segue as mesmas recomendações e procedimentos. Constitui diferencial favorável, a combinação de metodologias e ferramentas CASE que destinam-se a automatizar essas metodologias.

Ferramentas CASE podem ser consideradas como a automação da automação, pois destinam-se a automatizar a tarefa de desenvolvimento e manutenção de software, cujo propósito do mesmo, é automatizar alguma tarefa anteriormente executada manualmente. A tecnologia CASE automatiza a elaboração dos diversos diagramas do software em construção, a definição do seu dicionário de dados, a interação entre o usuário e o computador através de sua interface e as interações entre o software e o meio externo em que está inserido. A partir dessa especificação completa do software, a ferramenta poderá gerar automaticamente o aplicativo, onde o software gerado será mais confiável e com menos erros, facilitando futuras manutenções, proporcionando uma redução no custo do mesmo.

Existem diversas ferramentas CASE disponíveis no mercado. Cabe à organização escolher a que melhor atende às suas metas estabelecidas e requisitos necessários. Essas ferramentas normalmente proporcionam a automatização de uma tarefa pontual, ou seja, destinam-se a prover recursos para uma fase do ciclo de vida do desenvolvimento de software. Nesse caso, pode ser necessária a combinação de diversas ferramentas num ambiente totalmente integrado para atender às necessidades da organização.

A tarefa de escolher a ferramenta ou conjunto delas que melhor atende à organização não é fácil, pois envolve alguns fatores determinantes como o comprometimento da gerência com a adoção de CASE, uma equipe competente para analisar essas ferramentas e a aceitação da equipe de desenvolvimento em adotar a tecnologia CASE para a automação das suas atividades, pois a princípio poderá haver resistência quanto ao seu uso.

Ao longo do tempo, foram propostas algumas formas de avaliar e selecionar ferramentas CASE. Pode ser através da elaboração de questionários acerca das características das ferramentas, por meio de conversas com outras organizações que já fazem uso da tecnologia, apresentações de fornecedores de CASE, ou mesmo o desenvolvimento de projetos piloto para avaliar e selecionar a ferramenta mais adequada para a organização.

A norma ISO/IEC 14102 – Avaliação e seleção de ferramentas CASE, propõe um roteiro a seguir, com o objetivo de facilitar essa tarefa. Através de seus quatro processos, iniciação, estruturação, avaliação e seleção, instrui a organização sobre a melhor forma de conduzir uma correta avaliação e seleção de ferramentas CASE.

O software desenvolvido, baseado na norma ISO/IEC 14102, destina-se a automatizar algumas tarefas referentes aos processos da mesma. Este software foi utilizado durante o primeiro semestre de 1999 pelos alunos da disciplina de Engenharia de Software da FURB e pelo autor desse trabalho, na avaliação de algumas ferramentas, comprovando sua aplicabilidade e relevância. Também apresenta seu lado didático ao proporcionar aos alunos que o utilizaram, um software que facilita a avaliação de ferramentas CASE bem como um contato com a norma ISO/IEC 14102 – Avaliação e seleção de ferramentas CASE. O software facilita a adoção da norma a medida que automatiza tarefas importantes nos seus processos como por exemplo, a montagem do questionário de avaliação, o registro das notas atribuídas às subcaracterísticas das ferramentas avaliadas, o algoritmo de acumulação da pontuação obtida e posterior a esse, a emissão da relação com as ferramentas candidatas com as respectivas pontuações para que se processe a seleção da ferramenta que atende aos requisitos da organização.

A organização que irá avaliar e selecionar ferramentas CASE, poderá decidir pela seleção de um conjunto restrito de características e subcaracterísticas de ferramentas CASE para avaliar, de acordo com suas necessidades e seus métodos de desenvolvimento e

manutenção de software. O software possibilita essa seleção de itens relevantes à organização, à medida que permite eliminar do questionário de avaliação, itens que tem pouca ou nenhuma importância para a organização.

5.2 SUGESTÕES

Para fins de melhoramento do software, sugere-se o desenvolvimento do planejamento e controle do projeto que pode contar com a definição de um cronograma, seguindo o modelo de planejamento e controle de projetos adotados pela organização.

Sugere-se também no software, possibilitar a inclusão de comentários para cada nota atribuída às subcaracterísticas de ferramentas CASE avaliadas.

Muitas das subcaracterísticas de ferramentas CASE apresentam alto grau de subjetividade. Surge a necessidade de avaliar tais subcaracterísticas para torná-las mais objetivas. Sugere-se a criação de *check-lists* para cada uma dessas subcaracterísticas com o objetivo de facilitar a avaliação, reduzindo drasticamente ou eliminando por completo a subjetividade presente em cada avaliador. A partir da verificação dos itens dos *check-lists*, será atribuída objetivamente uma nota à cada subcaracterística avaliada.

Outra sugestão, é que se faça um mapeamento nas organizações que desenvolvem software, quanto ao uso de metodologias de desenvolvimento e ferramentas CASE, a fim de melhor conhecer a realidade da região no que se refere à produção de software.

REFERÊNCIAS BIBLIOGRÁFICAS

- [BLA94] BLANC, Louis A Le; KORN, Willard M. **A phased approach to the evaluation and selection of CASE tools**. Information and Software Technology, volume 36, número 5, páginas 267-273, 1994.
- [CHI93] CHIKOFSKI, Elliot J. **Computer-Aided Software Engineering (CASE)**; 2nd ed. p. cm. IEEE Computer Society Press technology series, 1993.
- [CLU89] McCLURE, Carma. **CASE is software automation**. New Jersey : Prentice-Hall, 1989.
- [COO94] COOKE, Daniel E.; BOONE, Greg. **The impact of CASE technology on software processes – Establishing the context of continuous improvement for technology transfer**. Singapore : World Scientific – Utopia Press, 1994.
- [FEL88] FELICIANO Neto, Acácio; HIGA, Wilson; FURLAN, José Davi. **Engenharia da informação : metodologia, técnicas e ferramentas**. São Paulo : McGraw-Hill, 1988.
- [FIS90] FISCHER, Alan S. **CASE – Utilização de ferramentas para desenvolvimento de software**. Tradução Info-Rio. Rio de Janeiro : Campus, 1990.
- [FOU94] FOURNIER, Roger. **Guia prático para o desenvolvimento e manutenção de sistemas estruturados**. Tradução Flávio Deny Steffen; revisão técnica Silvio Carmo Palmieri. São Paulo : Makron Books, 1994.
- [FUG93] FUGGETTA, Alfonso. **A classification of CASE technology**. COMPUTER. IEEE Computer Society, december, 1993.
- [ISO98] ISO/IEC 14102. **Norma para avaliação e seleção de ferramentas CASE ISO/IEC 14102 - versão inicial da tradução para o idioma Português**. Rio de Janeiro : ABNT, 1998.
- [JOA93] JOÃO, Belmiro do Nascimento. **Metodologias de desenvolvimento de sistemas**. São Paulo : Érica, 1993.
- [MAR83] MARTIN, James; McCLURE, Carma. **Software maintenance – the problem and its solutions**. New Jersey : Prentice-Hall, 1983.

- [MAR91] MARTIN, James; McCLURE, Carma. **Técnicas estruturadas e CASE**. Tradução Lúcia Faria Silva; revisão técnica Ronald Stevis Cassiolato. São Paulo : Makron Books, McGraw-Hill, 1991.
- [MAR95] MARTIN, James; ODELL, James J. **Análise e projeto orientados a objeto**. Tradução José Carlos Barbosa dos Santos; revisão técnica Ronald Stevis Cassiolato. São Paulo : Makron Books, 1995.
- [MAY98] MAYER, Roberto Carlos. **Brasil Software – relatório de pesquisa de mercado sobre o desenvolvimento de software no Brasil**; 8ª ed. - março de 1998. São Paulo : Mayer & Bunge Informática, 1998.
- [NBR96] NBR 13596/1996. **Tecnologia de Informação – Avaliação de produto de software – Características de qualidade e diretrizes para o seu uso**. Rio de Janeiro : ABNT, 1996.
- [NBR97] NBR ISO/IEC 12207/1997. **Tecnologia de Informação – Processos de ciclo de vida de software**. Rio de Janeiro : ABNT, 1997.
- [PRE95] PRESSMAN, Roger S. **Engenharia de Software**. Tradução José Carlos Barbosa dos Santos; revisão técnica José Carlos Maldonado, Paulo Cesar Masiero, Rosely Sanches. São Paulo : Makron Books, 1995.
- [RUT96] RUTZEN, Marcelo Fernando. **Um protótipo de editor gráfico para um meta-case**. Trabalho de conclusão de curso (Bacharelado em Ciências da Computação). Centro de Ciências Exatas e Naturais. Blumenau : FURB, 1996.
- [SIL98] SILVA, Luís Carlos. **Há ferramentas sérias para desenvolvedores sérios no mercado ?**. Developers´ Magazine, ano II, número 22, junho 1998, páginas 20-22. Rio de Janeiro : Axcel Books do Brasil Editora Ltda., 1998.
- [WEB94] WEBER, Kival Chaves; MILLET, Paulo Barreira; BRANDÃO, Dorgival Filho. **Qualidade e produtividade em software**. Brasília : QA&T Consultores Associados, 1994.