



Universidade Federal de Pernambuco
Centro de Informática



Curso Ciência da Computação
Turma 2004.2



Um Processo de Avaliação da Portabilidade de Unidades de Software

Autor:

Marcílio José Albuquerque Gomes Filho (mjagf@cin.ufpe.br)

Orientador:

Prof.º Alexandre Marcos Lins de Vasconcelos

Recife, Março 2005

Marcílio José Albuquerque Gomes Filho

Um Processo de Avaliação da Portabilidade de Unidades de Software

*Trabalho apresentado à Coordenação da
Graduação em Ciência da Computação do
Centro de Informática da Universidade
Federal de Pernambuco, como requisito
parcial para a obtenção do título de Bacharel
em Ciência da Computação.*

Orientador:

Prof. Dr. Alexandre Marcos Lins de Vasconcelos

Recife, março de 2005

Agradecimentos

Aos meus pais, por terem construído, com muita luta, uma família onde sempre encontro forças para continuar.

A minha irmã, que pacientemente suportou durante esses quatro anos e meio todo o meu stress.

Ao meu orientador, professor Alexandre Vasconcelos, por ter aceitado a difícil tarefa de me orientar.

A Apply Solutions, empresa que me acolheu e que disponibilizou informações para a realização deste trabalho.

Aos meus amigos.

A todos aqueles que, de forma direta ou não, contribuíram para que tivesse êxito.

Resumo

O presente trabalho tem por objetivo apresentar uma proposta de um processo de avaliação da portabilidade de unidades de software aos moldes das normas internacionais ISO/IEC 9126 e 14598. As subcaracterísticas de portabilidade consideradas são aquelas descritas na ISO/IEC 9126-1 e o processo de avaliação segue as recomendações da família de normas ISO/IEC 14598. Esse processo de avaliação está inserido no contexto do Laboratório de Avaliação de Produtos de Software, o LAPS. Este trabalho também trás um breve estudo sobre portabilidade e, ao final, apresenta os resultados da aplicação do modelo proposto na avaliação de um produto de software.

Palavras-chave: Qualidade de Software, Avaliação de Produtos de Software, Portabilidade.

Sumário

AGRADECIMENTOS	4
1 INTRODUÇÃO	1
1.1 Motivação	1
1.2 Objetivo do Trabalho.....	3
1.3 Estrutura do Trabalho	4
2 QUALIDADE E AVALIAÇÃO DE PRODUTOS DE SOFTWARE.....	5
2.1 A Família ISO/IEC 9126	6
2.2 A Família ISO/IEC 14598	10
2.3 O Projeto LAPS	13
2.4 Considerações Finais.....	16
3 PORTABILIDADE.....	17
3.1 Introdução	17
3.2 Definições.....	18
3.3 Considerações Finais.....	22
4 PROCESSO DE AVALIAÇÃO DA PORTABILIDADE DE UNIDADES DE SOFTWARE.....	23
4.1 Introdução	23
4.2 Selecionar Métricas e Seus Níveis de Pontuação	24
4.2.1 Métricas Internas de Adaptabilidade	25
4.2.2 Métricas Internas de Capacidade para ser Instalado	26
4.2.3 Métrica Interna de Coexistência	27
4.2.4 Métrica Interna de Capacidade para Substituir	27
4.2.5 Métricas Externas de Adaptabilidade	28
4.2.6 Métrica Externa de Capacidade para ser Instalado.....	28

4.2.7	Métrica Externa de Coexistência	29
4.2.8	Métrica Externa de Capacidade para Substituir	29
4.3	Estabelecer Níveis de Pontuação para as Métricas	29
4.4	Obter e Julgar os Valores Obtidos	31
4.4.1	Identificar as Informações Disponíveis	32
4.4.2	Identificar Dependências	33
4.4.3	Selecionar o Método de Avaliação	34
4.4.4	Propor Novos Ambientes	35
4.4.5	Executar a Avaliação.....	36
4.4.6	Sintetizar os Resultados	37
4.5	Considerações Finais.....	38
5	ESTUDO DE CASO	39
5.1	Escopo da Avaliação	39
5.2	A Avaliação	40
5.3	Resultados	41
5.4	Considerações Finais.....	42
6	CONCLUSÃO	44
6.1	Principais Contribuições	45
6.2	Dificuldades Encontradas	45
6.3	Trabalhos Futuros	46
	REFERÊNCIAS	48
	APÊNDICE A – CHECKLIST DO PROCESSO DE PORTABILIDADE.....	51
	APÊNDICE B – QUESTIONÁRIO SOBRE PORTABILIDADE	53
	APÊNDICE C – RELATÓRIO DE DEPENDÊNCIAS.....	55
	APÊNDICE D – RELATÓRIO DE DESCRIÇÃO DOS AMBIENTES	57

Índice de Figuras

Figura 1.1 – Relação entre as normas ISO/IEC 9126 e ISO/IEC 14598.....	2
Figura 2.1 – Modelo de qualidade para qualidade externa e interna.	7
Figura 2.2 – Processo de avaliação definido pela ISO/IEC 14598.	11
Figura 2.3 – Visão geral dos módulos de avaliação do LAPS.	14
Figura 4.1 – Processo de Avaliação da Portabilidade.....	32

Índice de Tabelas

Tabela 4.1 – Escala para métricas internas.....	30
Tabela 4.2 – Escala para métricas externas.....	30
Tabela 4.3 – Resumo da atividade Identificar dependências.	34
Tabela 4.4 – Resumo da atividade Selecionar o método de avaliação.....	35
Tabela 4.5 – Resumo da atividade Propor novos ambientes.	36
Tabela 4.6 – Resumo da atividade Executar a avaliação.....	37
Tabela 4.7 – Resumo da atividade Sintetizar os resultados.....	37
Tabela 5.1 – Resultados da aplicação das métricas.	41

1 Introdução

Este capítulo aborda sobre a motivação e as principais fontes de informação deste estudo, como também apresenta a estrutura deste trabalho, procurando dar uma visão de alto nível sobre os assuntos que serão abordados neste documento.

1.1 Motivação

A expansão do mercado de tecnologia da informação vem sendo alavancada pela crescente necessidade das áreas industrial e de serviços em automatizar processos e aumentar a produção. Nesse contexto, os produtos de software têm se tornado cada vez mais importantes e, em alguns casos, essenciais, para que as empresas desses setores alcancem suas metas.

Esse crescimento do mercado de TI trás consigo o aumento da concorrência entre as empresas produtoras de software, as quais precisam buscar diferenciais para agregar valor a seus produtos e, assim, expandir seu mercado consumidor.

Um dos fatores, talvez o principal, que vem se mostrando eficiente em diferenciar empresas produtoras de software é a implantação de um processo de garantia da qualidade de software.

A avaliação da qualidade do produto de software tem sido uma das formas empregadas por organizações que produzem ou adquirem software para obtenção de maior qualidade nestes produtos [2].

Para que a avaliação seja mais efetiva, é importante que se utilize um modelo de qualidade que permita estabelecer e avaliar requisitos de

qualidade e também que o processo de avaliação seja bem definido e estruturado [2].

Este trabalho será focado no modelo de qualidade descrito na ISO/IEC 9126-1 e no processo de avaliação descrito na família de normas ISO/IEC 14598.

A figura 1.1 apresenta a relação entre essas duas famílias de normas da ISO/IEC.

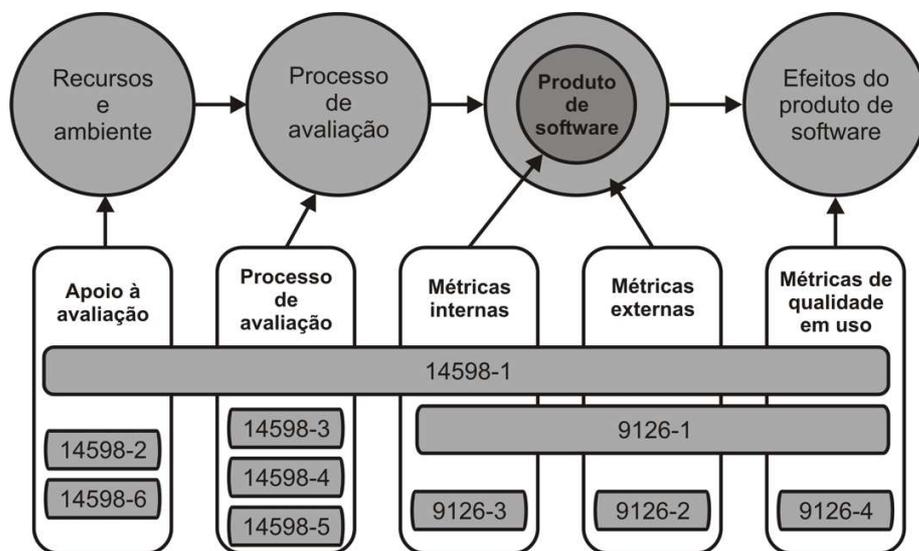


Figura 1.1 – Relação entre as normas ISO/IEC 9126 e ISO/IEC 14598.

Observando-se mais cuidadosamente esta figura, pode-se perceber como se dá esse relacionamento: a ISO/IEC 14598-1 apresenta um processo de avaliação de produtos de software, que tem como pré-requisito, a definição de um modelo de qualidade, modelo este definido na ISO/IEC 9126-1. Este mesmo processo sugere como se deve proceder para coletar e registrar os valores dos atributos de qualidade a serem avaliados, atributos esses sugeridos pelas normas ISO/IEC 9126-2, ISO/IEC 9126-3 e ISO/IEC 9126-4.

O modelo de qualidade da ISO/IEC 9126-1 é representado por um desdobramento hierárquico das características de qualidade do produto de software, estando bem definido nos seus dois primeiros níveis (características e subcaracterísticas) e deixando o terceiro nível de desdobramento (atributos) a critério do usuário do modelo [2].

Segundo a ISO/IEC 14598-5, o objetivo principal dos processos de avaliação é promover a repetibilidade, a reprodutibilidade, a imparcialidade e a objetividade.

- *Repetibilidade*: avaliações repetidas do mesmo produto, com a mesma especificação de avaliação, pelo mesmo avaliador, produzam resultados que possam ser aceitos como idênticos.
- *Reprodutibilidade*: avaliações do mesmo produto, com a mesma especificação de avaliação, por avaliadores diferentes, produzam resultados que possam ser aceitos como idênticos.
- *Imparcialidade*: avaliações não devem ser tendenciosas com relação a algum resultado particular.
- *Objetividade*: os resultados das avaliações devem ser baseados em fatos, isto é, desprovidos de sentimentos ou opiniões do avaliador.

1.2 Objetivo do Trabalho

O objetivo deste trabalho é definir um processo de avaliação de unidades de software baseado na família ISO/IEC 14598, focado na característica portabilidade e suas subcaracterísticas (adaptabilidade, capacidade para ser instalado, coexistência e capacidade para substituir) descritas na ISO/IEC 9126-1. A característica portabilidade foi escolhida

porque é muito importante no desenvolvimento de sistemas, pois estende o valor e a vida útil das unidades de software [5].

1.3 Estrutura do Trabalho

Este capítulo apresentou uma introdução fornecendo conteúdo para as explicações dos capítulos seguintes.

A distribuição do conteúdo pelos demais capítulos será especificada a seguir:

O capítulo 2 é dedicado à qualidade de produto de software e às normas da ISO/IEC que definem as características de qualidade de produtos de software e processos de avaliação dessas características. Também trás uma seção sobre laboratórios de avaliação de produtos de software.

O capítulo 3 apresenta um breve estudo sobre portabilidade, trazendo um pouco de história e conceitos associados a essa característica de qualidade de produtos de software.

O capítulo 4 apresenta o processo de avaliação da portabilidade de produtos de software proposto neste estudo, com todo o passo a passo necessário para a realização de avaliações.

O capítulo 5 apresenta os resultados da avaliação da qualidade da portabilidade de um produto de software aos moldes do processo de avaliação definido no capítulo anterior.

Por último, são apresentadas conclusões e considerações finais sobre o processo de avaliação da portabilidade de produtos de software definido neste estudo.

2 Qualidade e Avaliação de Produtos de Software

Há quem diga que implantar um processo de garantia da qualidade de software não é uma opção a ser estudada, mas parte de uma estratégia de sobrevivência em um mercado cada vez mais exigente e competitivo [1].

A qualidade de software é um processo sistemático que focaliza todas as etapas e artefatos produzidos com o objetivo de garantir a conformidade de processos e produtos, prevenindo e eliminando defeitos [1]. Ela é abordada sob dois aspectos ou dimensões: qualidade do processo e qualidade do produto. A qualidade do processo tem como foco principal garantir a qualidade de cada etapa dos processos definidos pela engenharia de software (especificação, projeto, implementação, validação e evolução) e a qualidade do produto tem como objetivo principal garantir a qualidade dos produtos tecnológicos gerados durante e depois do ciclo de desenvolvimento de software. Este trabalho será focado na dimensão de qualidade do produto, mais precisamente na avaliação da qualidade da portabilidade de unidades de software.

Unidade de Software é um termo empregado de forma ampla, e representa diversas entidades de um sistema computacional [4], como por exemplo, programas, componentes e API's.

A avaliação de produtos de software, com o objetivo de satisfazer as necessidades de qualidade de software, deve ser encarada como um dos processos no ciclo de vida de desenvolvimento de software [7]. O objetivo principal da avaliação de produtos de software é fornecer resultados

quantitativos sobre a qualidade do produto de software que sejam compreensíveis, aceitáveis e confiáveis por quaisquer das partes interessadas (desenvolvedores, fornecedores e adquirentes) [3], visando apoiar diretamente o desenvolvimento e a aquisição de produtos de software.

A primeira etapa na avaliação de produtos de software consiste em selecionar as características de qualidade relevantes, utilizando um modelo de qualidade que desdobre a qualidade de produtos de software em diferentes características [8].

A segunda etapa consiste em definir um processo de avaliação que esteja de acordo com regras e critérios internacionais, a fim de garantir que o resultado final esteja em concordância com os mais exigentes critérios de avaliação de produtos de software.

Visando atender aos requisitos acima citados, em relação ao modelo de qualidade e ao processo de avaliação, este estudo está embasado no modelo de qualidade definido pela família de normas ISO/IEC 9126 e pelo modelo de avaliação definido pela família de normas ISO/IEC 14598, os quais são internacionalmente reconhecidos e aceitos, bem como está inserido no contexto do Laboratório de Avaliação de Produtos de Software, o LAPS, o qual também segue essas normas internacionais.

2.1 A Família ISO/IEC 9126

Esta família de normas é formada por quatro documentos, a saber: ISO/IEC 9126-1, ISO/IEC 9126-2, ISO/IEC 9126-3 e ISO/IEC 9126-4. Em conjunto, descrevem um modelo de qualidade do produto de software composto por duas partes complementares. A primeira parte, chamada

qualidade interna e externa, está contemplada nas normas ISO/IEC 9126-1, ISO/IEC 9126-2 e ISO/IEC 9126-3 e especifica um conjunto de seis características de qualidade interna e externa, um conjunto de subcaracterísticas para cada característica e métricas para a avaliação destas. A segunda parte, conhecida por qualidade em uso, especifica quatro características de qualidade em uso e está contemplada na ISO/IEC 9126-4. Neste trabalho será abordada apenas a primeira parte, qualidade interna e externa, mais precisamente na característica portabilidade e suas subcaracterísticas.

A figura 2.1 apresenta as características e subcaracterísticas definidas na primeira parte do modelo de qualidade do produto de software definido pela família de normas ISO/IEC 9126.

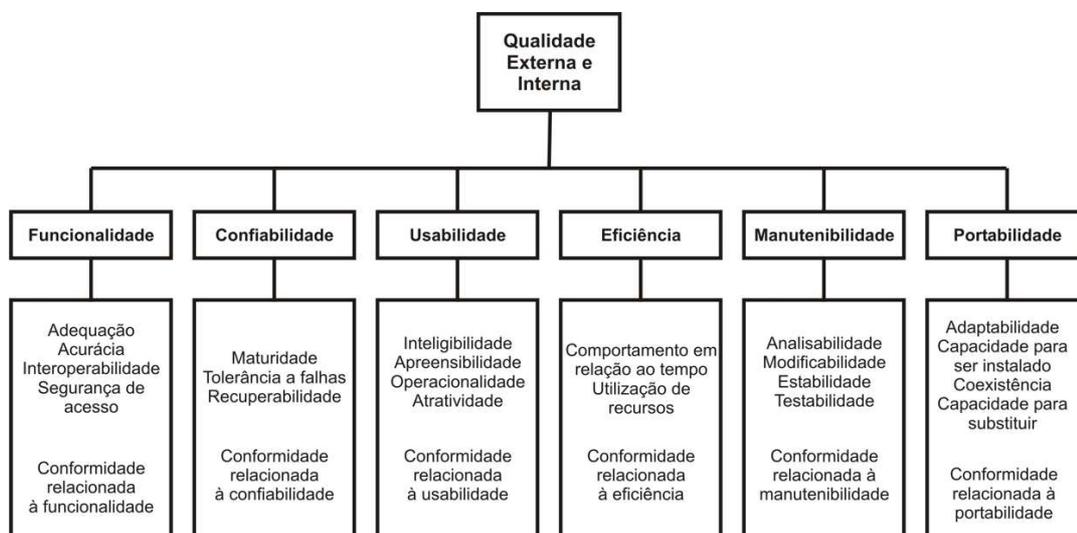


Figura 2.1 - Modelo de qualidade para qualidade externa e interna.

Essa estruturação da qualidade externa e interna de produtos de software, em características e subcaracterísticas, é aplicável a todo tipo de software, desde programas de computador a dados contidos em *firmware*, e fornece terminologia consistente para se tratar de qualidade do produto

de software, bem como fornece uma estrutura para especificar requisitos de qualidade de software e realizar comparações entre produtos de software [7].

As características e suas respectivas subcaracterísticas foram definidas de forma a deixar claro o escopo de cada uma delas, a fim de eliminar ambigüidades que possam surgir a partir de diferentes interpretações.

A seguir serão apresentadas as descrições das seis características definidas.

- *Funcionalidade*: capacidade do produto de software de prover funções que atendam às necessidades explícitas e implícitas, quando o software estiver sendo utilizado sob condições especificadas [7].
- *Confiabilidade*: capacidade do produto de software de manter um nível de desempenho especificado, quando usado em condições especificadas [7].
- *Usabilidade*: capacidade do produto de software de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições especificadas [7].
- *Eficiência*: capacidade do produto de software de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob condições especificadas [7].
- *Manutenibilidade*: capacidade do produto de software de ser modificado. As modificações podem incluir correções, melhorias ou

adaptações do software devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais [7].

- *Portabilidade*: capacidade do produto de software de ser transferido de um ambiente para outro [7].

Como mencionado anteriormente, cada uma dessas características pode ser decomposta em diversas subcaracterísticas, as quais constituem um menor nível de abstração. Contudo, devido ao foco desta monografia, serão apresentadas, apenas, as subcaracterísticas de portabilidade.

- *Adaptabilidade*: capacidade do produto de software de ser adaptado para diferentes ambientes especificados, sem necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado [7].
- *Capacidade para ser instalado*: capacidade do produto de software para ser instalado em um ambiente especificado [7].
- *Coexistência*: capacidade do produto de software de coexistir com outros produtos de software independentes, em um ambiente comum, compartilhando recursos comuns [7].
- *Capacidade para substituir*: capacidade do produto de software de ser usado em substituição a outro produto de software especificado, com o mesmo propósito e no mesmo ambiente [7].
- *Conformidade relacionada à portabilidade*: capacidade do produto de software de estar de acordo com normas ou convenções relacionadas à portabilidade [7].

Como pode ser percebido, o modelo de qualidade definido pela família de normas ISO/IEC 9126 é representado por um desdobramento hierárquico das características de qualidade de produto de software, estando bem definido nos seus dois primeiros níveis de abstração (características e subcaracterísticas) [2]. Esta forma de definir as características não permite sua medição direta, principalmente devido ao atual estado da arte em medição de software [3]. O terceiro nível de desdobramento (atributos) deve ser definido a critério do usuário do modelo [2] para só então estabelecer métricas que se correlacionem às características do produto de software.

As normas ISO/IEC 9126-2 e ISO/IEC 9126-3 apresentam algumas sugestões de métricas externas e internas, respectivamente. As métricas externas utilizam medidas de um produto de software derivadas de medidas de comportamento do sistema do qual o software é uma parte. Já as métricas internas podem ser aplicadas a um produto de software não executável durante o projeto e a codificação [7].

2.2 A Família ISO/IEC 14598

Esta família de normas é composta por seis documentos, a saber: ISO/IEC 14598-1, ISO/IEC 14598-2, ISO/IEC 14598-3, ISO/IEC 14598-4, ISO/IEC 14598-5 e ISO/IEC 14598-6. Em conjunto, apresentam uma visão geral do processo de avaliação de produto de software e fornecem orientações e requisitos para as avaliações [8], garantindo que os resultados obtidos possam ser utilizados por gerentes, desenvolvedores e mantenedores para medir a aderência aos requisitos e para fazer melhorias onde necessárias.

As normas ISO/IEC 14598-2 e a ISO/IEC 14598-6 são relacionadas ao suporte e gestão da avaliação em nível corporativo ou departamental, oferecendo informações como: quais atividades devem ser executadas, em que ordem e orientação para a documentação de módulos de avaliação. Já as normas ISO/IEC 14598-3, ISO/IEC 14598-4 e ISO/IEC 14598-5 fornecem requisitos e orientações para o processo de avaliação em nível de projeto em três diferentes situações: desenvolvimento, aquisição e avaliação independente, respectivamente.

A figura 3.1 mostra o processo de avaliação proposto pela família de normas ISO/IEC 14598.



Figura 2.2 - Processo de avaliação definido pela ISO/IEC 14598.

A seguir, serão apresentados os objetivos de cada um dos passos do processo de avaliação definido pela família de normas ISO/IEC 14598.

- *Estabelecer o propósito da avaliação:* o objetivo final é assegurar que o produto forneça a qualidade requerida.

- *Identificar tipos de produto(s) a serem avaliados:* o objetivo é identificar se o produto a ser avaliado é um produto intermediário do processo de desenvolvimento ou um produto final.
- *Especificar modelo de qualidade:* o objetivo é definir qual o modelo de qualidade que será adotado no processo de avaliação.
- *Selecionar métricas:* o objetivo é garantir que as métricas utilizadas no processo de avaliação possibilitem uma avaliação fácil e economicamente viável.
- *Estabelecer níveis de pontuação para as métricas:* o objetivo é criar as escalas de valores das métricas selecionadas.
- *Estabelecer critérios para julgamento:* o objetivo é definir os critérios que serão utilizados durante a interpretação dos resultados das medições.
- *Produzir o plano de avaliação:* o objetivo é criar o plano de avaliação, o qual contém a descrição de todos os métodos de avaliação e o cronograma da avaliação.
- *Obter as medidas:* o objetivo é executar a avaliação e coletar os resultados para posterior análise.
- *Comparar com critérios:* o objetivo é comparar os valores obtidos durante o processo de avaliação com os critérios predefinidos.
- *Julgar os resultados:* o objetivo é resumir as informações extraídas do processo de avaliação de software para declarar o quanto o produto de software atende aos requisitos de qualidade.

2.3 O Projeto LAPS

Uma das formas de implementar um processo de avaliação de produtos de software que possua um modelo de qualidade estruturado e seja bem definido é através da criação de um laboratório de avaliação de produtos de software. Esse laboratório pode ser uma área interna de uma empresa desenvolvedora de software ou pode ser uma empresa por si só, a qual presta serviços de avaliação.

O presente trabalho está inserido no contexto de um laboratório de avaliação, o Laboratório de Avaliação de Produtos de Software, ou LAPS, que se encontra em processo de afirmação no mercado de TI de Recife. Este laboratório traz a proposta de dividir a avaliação de produtos de software em módulos de serviços diferenciados, onde esses módulos segmentem a avaliação da qualidade, permitindo aos clientes identificar e combinar, por grau de importância, conforme sua necessidade, a análise a ser realizada, permitindo que a avaliação da qualidade possa ser feita de forma parcial (alguns módulos) ou total (todos os módulos) [6].

Devido ao ecossistema local de empresas de TI apresentar uma grande parcela de empresas que adotaram o paradigma de orientação a objetos no desenvolvimento de soluções, os processos de avaliação de produtos de software utilizados pelo LAPS são direcionados para software implementado em linguagens orientadas a objetos como Java, C++ e C#.

A metodologia adotada para identificação dos módulos foi baseada na ISO/IEC 9126 e o processo de avaliação está em conformidade com a ISO/IEC 14598-1.

A figura 2.3 fornece uma visão geral dos módulos disponíveis.

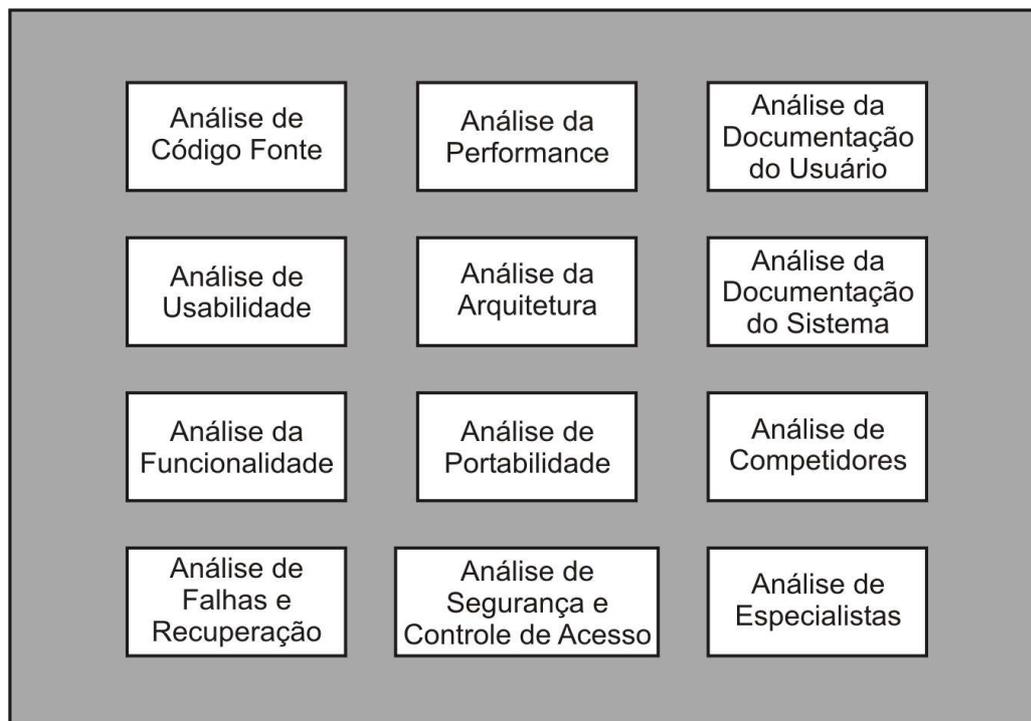


Figura 2.3 – Visão geral dos módulos de avaliação do LAPS.

O módulo de avaliação da arquitetura tem como objetivo avaliar se a arquitetura projetada para o sistema é capaz de atender as funcionalidades desejadas e respeita requisitos não funcionais de modificabilidade e analisabilidade.

O módulo de avaliação da documentação do sistema visa medir a qualidade dos documentos gerados no processo de desenvolvimento do sistema.

O módulo de avaliação da funcionalidade tem por objetivo avaliar o conjunto de funções especificadas e suas propriedades verificando seu grau de qualidade.

O módulo de avaliação da portabilidade objetiva a inspeção da facilidade na qual o software pode ser transferido de um sistema computacional ou ambiente para outro.

O módulo de avaliação da usabilidade objetiva a inspeção da facilidade, clareza e intuitividade dos softwares, almejando a melhor compreensão das funcionalidades do sistema pelo usuário.

O módulo de avaliação do código fonte visa verificar o código fonte de um produto objetivando medir o grau de qualidade do mesmo.

O módulo de avaliação de desempenho visa estimar tempo e recursos consumidos, descobrir potenciais pontos de gargalo e prever o desempenho em um ambiente real.

O módulo de avaliação da documentação do usuário tem como objetivo avaliar se a documentação do usuário está clara no ensino e esclarecer dúvidas que possam surgir durante a utilização do produto.

O módulo de avaliação de falhas e recuperação visa determinar a natureza exata e a localização de erros a fim de que se possa identificar e corrigir erros semelhantes e iniciar ação corretiva para prevenção de futuras ocorrências de erros semelhantes.

O módulo de avaliação de controle de acesso e proteção dos dados visa a identificar deficiências no software no que diz respeito ao acesso de usuários não-identificados ou erroneamente identificados ao sistema, bem como a verificação da possibilidade de obtenção, por quaisquer meios ilícitos, de informações consideradas confidenciais ao sistema.

O módulo de avaliação de competidores tem como finalidade avaliar a posição do produto no Mercado e em relação aos seus concorrentes.

O módulo de avaliação de especialista objetiva acompanhar e sistematizar a avaliação do produto por especialistas da área, sem com isso, colocar entraves de aspecto tecnológico.

Este trabalho constitui o alicerce para a criação do módulo de avaliação da portabilidade, pois assim como a metodologia adotada pelo LAPS, este estudo está direcionado pelas normas ISO/IEC 9126 e ISO/IEC 14598.

2.4 Considerações Finais

Este capítulo trouxe uma breve discussão sobre qualidade e avaliação de produtos de software. Foram apresentadas as principais normas internacionais que se relacionam com qualidade e avaliação e também houve uma breve explanação sobre laboratórios de avaliação de produtos de software.

O próximo capítulo trará um estudo sobre portabilidade, de acordo com as normas da ISO/IEC e de estudos realizados por pesquisadores desde a década de 1990.

3 Portabilidade

Unidades de software nunca executam sozinhas, mas sempre como parte de um sistema maior [7], ou seja, no contexto de um ambiente computacional, interagindo com hardware e software. Esta interação com o ambiente se dá através de uma coleção de interfaces com a CPU, a memória, o sistema operacional e outros sistemas.

3.1 Introdução

O termo ambiente é um termo amplo e refere-se tanto à plataforma de hardware quanto ao conjunto de softwares com os quais a unidade de software deve interagir. Esses ambientes podem apresentar diversas variações em sua configuração, tanto no nível de hardware quanto no nível de software, possibilitando uma grande quantidade de combinações possíveis. São exemplos de variações no nível de hardware: a ordem em que os bits e bytes são armazenados na memória, a quantidade de bits utilizada para representar valores inteiros e endereços de memória, os valores mínimo e máximo de números inteiros e a forma de representar os números de ponto flutuante. Já as variações a no nível de software são ainda maiores, pois além do próprio sistema operacional, com o qual a unidade de software deve interagir, qualquer outro software com o qual a unidade de software troque informações ou compartilhe recursos apresenta a possibilidade de mudança entre um ambiente e outro.

Avaliando o quadro apresentado acima, fica claro que desenvolver software que tenha a capacidade de ser executado de forma satisfatória em ambientes diferentes é um requisito mais que desejável e também

fundamental para estender a vida útil da unidade de software, como também agregar valor ao produto de software final [5].

3.2 Definições

Visando o objetivo especificado no parágrafo anterior, durante o ciclo de desenvolvimento de unidades de software, alguns atributos de qualidade devem ser considerados para permitir esta flexibilidade aos produtos de software. Segundo a ISO/IEC 9126-1 [7], o conjunto de atributos que evidenciam a capacidade do software ser transferido de um ambiente para outro é chamado Portabilidade. Essa mesma norma aponta a portabilidade como uma das seis características de qualidade de software descritas no modelo de qualidade definido por ela, modelo este utilizado como referência para este estudo e apresentado na seção 2.1, e também apresenta quatro subcaracterísticas para portabilidade, a saber:

- *Adaptabilidade*: capacidade do produto de software de ser adaptado para diferentes ambientes especificados, sem necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado.
- *Capacidade para ser instalado*: capacidade do produto de software para ser instalado em um ambiente especificado.
- *Coexistência*: capacidade do produto de software de coexistir com outros produtos de software independentes, em um ambiente comum, compartilhando recursos comuns.
- *Capacidade para substituir*: capacidade do produto de software de ser usado em substituição a outro produto de software especificado, com o mesmo propósito e no mesmo ambiente.

Cada uma dessas características pode ser desmembrada em atributos, a fim de facilitar sua verificação e medição, como também está mencionado na ISO/IEC 9126-1.

Segundo Mooney, em seu artigo publicado pela IEEE em 1990 [9], uma unidade de software é portátil, dentro de uma classe de ambientes, se o esforço necessário para transportá-la e adaptá-la para um novo ambiente, pertencente à classe considerada, é menor que o esforço de desenvolvê-la novamente. Esta relação entre o esforço para adaptar e o esforço para refazer caracteriza o grau de portabilidade do software. Assim, dizer que um software apresenta um elevado grau de portabilidade é dizer que esta unidade de software requer pequenas adaptações para executar num dado conjunto de ambientes. Também fica evidenciado que dizer que um software é ou não portátil é incorreto, pois a portabilidade não é uma característica dualista (sim ou não), mas sim gradativa.

Dessa forma, fica evidenciado que o problema de tornar uma unidade de software portátil, para um dado conjunto de ambientes, está centrado na análise das interações desse software com as interfaces desses ambientes [9], bem como fica claro que a presença da característica portabilidade não é dualista (é portátil ou não), mas está presente de forma gradual.

Historicamente, visando solucionar o problema de portar unidades de software, houve uma divisão deste em duas etapas distintas: transporte e adaptação. A etapa de transporte consiste do transporte físico da unidade de software, através de mídias (CD's, DVD's, *Flash drives*) ou canais de comunicação (internet, *intranet*), de um ambiente para outro. Já

a fase de adaptação é caracterizada pela modificação da versão original do programa fonte da unidade de software, ou seja, qualquer mudança que seja necessária para a correta execução da unidade de software no ambiente alvo.

Atualmente, a etapa de transporte aparenta não ser mais um problema devido aos avanços tecnológicos das formas e meios de armazenamento digital. Contudo, a adaptação continua a oferecer desafios na busca por altos graus de portabilidade em unidades de software. Esses desafios consistem, entre outras coisas, em balancear esforço e tempo durante o ciclo de desenvolvimento do produto de software.

Estudos sobre a portabilidade de software, realizados principalmente na década de 1990, identificaram dois tipos de portabilidade: a portabilidade binária (*binary portability*) e a portabilidade de fonte (*source portability*). A portabilidade binária consiste na capacidade de uma versão binária da unidade de software (compilada) permitir ser transportada e executada em um novo ambiente sem qualquer modificação [9]. Já a portabilidade de fonte consiste na capacidade de o programa fonte da unidade de software permitir ser transportado e executado num novo ambiente necessitando apenas ser compilado novamente utilizando ferramentas disponíveis neste novo ambiente. Analisando esses tipos de portabilidade, torna-se claro que unidades de software que apresentam portabilidade binária possuem maior grau de portabilidade, pois o esforço para executá-la em um novo ambiente se resume ao esforço empregado no transporte da unidade de software de

um ambiente para outro, e como foi discutido anteriormente, este esforço é mínimo devido aos avanços tecnológicos atuais.

Esses mesmos estudos também identificaram várias estratégias para se obter unidades de software com alto grau de portabilidade. Essas estratégias vão desde a análise da classe de ambientes onde o software irá executar, até a escolha da linguagem de programação utilizada na implementação da unidade de software.

Em relação à classe de ambientes, foi mencionado que uma unidade de software interage com o ambiente através de interfaces. Dessa forma, estratégias de implementação de unidades de software que isolem esses pontos de comunicação e mantenham interfaces idênticas, ou muito parecidas, entre a unidade e o ambiente, a partir da aderência a padrões apropriados, obtêm sucesso na busca por altos níveis de portabilidade [9].

Considerando-se a escolha da linguagem de programação para a implementação da unidade de software, uma estratégia que se mostra eficiente é a opção por linguagens de programação que apresentem suporte à portabilidade através de suas características próprias e das ferramentas de suporte disponíveis (compiladores, interpretadores, máquinas virtuais). Essas linguagens oferecem aos desenvolvedores procedimentos, funções, bibliotecas e API's padronizadas para a comunicação com o ambiente, oferecem facilidades para a prática do desenvolvimento em módulos e também são padronizadas por alguma entidade internacional.

Contudo, para se obter unidades de software com elevado grau de portabilidade, as estratégias mencionadas acima não são suficientes. Há a

necessidade de que essa característica do software seja identificada nos estágios iniciais do ciclo de desenvolvimento do software a fim de permitir que análises e estudos sejam realizados visando identificar a classe de ambientes onde a unidade de software estará introduzida e, por conseguinte, as possíveis combinações de interfaces entre o produto de software e cada ambiente pertencente à classe identificada. Dessa forma, a característica de qualidade de produtos de software chamada portabilidade poderá ser alcançada de forma satisfatória.

3.3 Considerações Finais

Este capítulo apresentou um estudo sobre portabilidade, trazendo introdução, definições e técnicas descritas na literatura, evidenciando a necessidade de se considerar a portabilidade uma característica importante para unidades de software.

O próximo capítulo apresentará o objetivo maior deste trabalho, a proposta de um método de avaliação da portabilidade de unidades de software.

4 Processo de Avaliação da Portabilidade de Unidades de Software

Este capítulo apresenta o objetivo maior deste trabalho, como dito no capítulo de introdução: a definição de um processo de avaliação da portabilidade de unidades de software que esteja em conformidade com normas internacionais de qualidade de software.

4.1 Introdução

Segundo a seção 2 da ISO/IEC 14598-1 [8], uma especificação de avaliação de software está em conformidade com esta norma se utilizar os processos definidos na seção 6 da mesma e um modelo de qualidade apropriado e bem definido. Os processos definidos na seção 6 da ISO/IEC 14598-1 são aqueles apresentados de forma resumida pela figura 2.2.

O modelo de qualidade utilizado, como dito anteriormente, será aquele definido pela ISO/IEC 9126-1, o qual atende plenamente aos requisitos definidos pela seção 7.3 da ISO/IEC 14598-1.

No capítulo 2 foi dito que este estudo está inserido no contexto do Laboratório de Avaliação de Produtos de Software, o LAPS, e que o processo de avaliação do LAPS está em conformidade com a ISO/IEC 14598.

Essa informação é importante porque existem processos que possuem atividades de caráter gerencial, ou seja, são atividades que visam dar apoio à avaliação propriamente dita. Esses processos de apoio são devidamente abordados na metodologia de avaliação de produtos de software utilizada pelo LAPS e não serão considerados neste estudo. Como

exemplos dessas atividades podem ser citadas as atividades de contatar o requisitante da avaliação e coletar e registrar as informações iniciais sobre o produto a ser avaliado.

Dos processos apresentados pela figura 2.2, serão considerados neste trabalho os processos de Especificação e Execução da avaliação. O primeiro define a abrangência da avaliação e das medições a serem realizadas [3] e o segundo visa obter os resultados da execução das ações de medição e verificação do produto de software [3].

Essa visão dos processos de Projeto e Execução da avaliação está de acordo com a ISO/IEC 14598-5, que trata da avaliação de produtos de software sob a perspectiva do avaliador.

4.2 Selecionar Métricas e Seus Níveis de Pontuação

É importante que as medições de um produto de software possam ser feitas fácil e economicamente e que as medidas resultantes sejam fáceis de usar [8]. Também é importante que as métricas utilizadas sejam válidas e suficientemente precisas para permitir que sejam feitas comparações confiáveis entre produtos de software [7].

A ISO/IEC 9126-1 apresenta dois tipos de métricas: as métricas internas e as métricas externas. Como este trabalho visa estar em conformidades com essa norma, serão definidas métricas internas e externas que serão utilizadas durante o processo de avaliação da portabilidade de produtos de software.

As métricas internas são aquelas aplicáveis a um produto de software não executável, tais como uma especificação ou o programa-fonte, e medem propriedades intrínsecas desse produto através da análise

de suas propriedades estáticas. As medições obtidas são, normalmente, números ou frequências dessas propriedades estáticas [11].

As métricas externas representam a perspectiva externa da qualidade de software quando a unidade de software está em uso. As medidas utilizadas são derivadas de medidas do comportamento do ambiente do qual o produto avaliado é parte, através de testes e observações do produto em execução [12].

Como mencionado na seção 2.1, a ISO/IEC 9126-1 define um modelo de qualidade que apresenta características e subcaracterísticas de qualidade de produtos de software. Este modelo define portabilidade como uma das características e apresenta suas quatro subcaracterísticas: adaptabilidade, capacidade para ser instalado, coexistência e capacidade para substituir.

Também foi dito que cabe aos usuários do modelo de qualidade a tarefa de definir os atributos da unidade de software que serão mensurados a partir dessas subcaracterísticas. As métricas apresentadas a seguir estão divididas em métricas internas e externas e também são subdivididas pelas subcaracterísticas as quais se relacionam.

4.2.1 Métricas Internas de Adaptabilidade

- *Adaptabilidade ao ambiente de hardware*: esta métrica visa medir o quão adaptável é a unidade de software quando há mudanças no ambiente de hardware. Ela é definida pela fórmula $X = A/B$, onde X é o valor da adaptabilidade ao ambiente de hardware, A é o número de componentes que oferecem capacidade de executar em diferentes

ambientes de hardware e B é o número de componentes que deveriam oferecer essa capacidade.

- *Adaptabilidade ao ambiente de software*: esta métrica mede o quão adaptável é a unidade de software em relação a mudanças nos softwares que fazem parte do ambiente onde o produto estará inserido. Ela é definida pela fórmula $X = A/B$, onde X é o valor da adaptabilidade ao ambiente de software, A é o número de componentes que oferecem capacidade para executar em diferentes ambientes de software e B é o número de componentes que deveriam oferecer essa capacidade.
- *Grau de portabilidade*: esta métrica tem por objetivo medir a relação entre o esforço de modificar e o esforço de refazer cada componente da unidade de software que apresenta dependência com o ambiente de hardware ou software a fim de torná-lo portátil a este ambiente. Ela é a soma aritmética do grau de portabilidade de cada componente e é definida pela fórmula $X = 1 - (A/B)$, onde X é o grau de portabilidade do componente, A é o esforço necessário para adaptar o componente e B é o esforço requerido para refazer o componente.

4.2.2 Métricas Internas de Capacidade para ser Instalado

- *Esforço para instalar*: esta métrica mede o nível de esforço necessário para instalar o produto de software. Ela é definida pela fórmula $X = A/B$, onde X é o esforço para instalar, A é o número de passos de instalação automáticos e B é o número total de passos de instalação necessários.

- *Flexibilidade de instalação:* esta métrica visa medir quão flexível e customizável é o processo de instalação da unidade de software. Ela é definida pela fórmula $X = A/B$, onde X é a flexibilidade de instalação, A é o número de operações de instalação que admitem customização e B é o número total de operações de instalação necessárias.

4.2.3 Métrica Interna de Coexistência

- *Disponibilidade de coexistência:* esta métrica mede o quão flexível é o produto de software quando compartilha recursos com outros softwares sem causar impactos negativos a esses produtos. Ela é definida pela fórmula $X = A/B$, onde X é a medida de disponibilidade de coexistência, A é o número de produtos previamente definidos com os quais a unidade de software pode coexistir sem problemas e B é o número de produtos no ambiente de produção com os quais o software deve coexistir sem problemas.

4.2.4 Métrica Interna de Capacidade para Substituir

- *Consistência de funcionalidades:* esta métrica tem por objetivo medir o quão consistente é a unidade de software em relação ao produto que está sendo substituído. Ela é definida pela fórmula $X = A/B$, onde X é a medida de consistência de funcionalidades da unidade de software, A é o número de funcionalidades da unidade de software que produzem resultados semelhantes aos do software antigo e B é o número de funcionalidades do produto substituído.

4.2.5 Métricas Externas de Adaptabilidade

- *Adaptabilidade ao ambiente de hardware*: esta métrica avalia a capacidade de auto-adaptação da unidade de software ao ambiente de hardware, o que oferece facilidades ao usuário do produto. Ela é definida pela fórmula $X = 1 - (A/B)$, onde X é a capacidade de auto-adaptação da unidade de software a mudanças no ambiente de hardware, A é o número de operações que não puderam ser executadas ou apresentaram resultados insatisfatórios devido a essas mudanças e B é o número de operações que foram executadas.
- *Adaptabilidade ao ambiente de software*: esta métrica visa avaliar a capacidade de auto-adaptação da unidade de software ao ambiente de software, o que traz facilidades ao usuário do produto. Ela é definida pela fórmula $X = 1 - (A/B)$, onde X é a capacidade de auto-adaptação da unidade de software a mudanças no ambiente de software, A é o número de operações que não puderam ser executadas ou apresentaram resultados insatisfatórios devido a essas mudanças e B é o número de operações que foram executadas.

4.2.6 Métrica Externa de Capacidade para ser Instalado

- *Facilidade de instalação*: esta métrica tem por objetivo avaliar o quão simples é o processo de instalação da unidade de software. Ela é definida pela fórmula $X = A/B$, onde X é a medida da facilidade de instalação, A é o número de intervenções do usuário no processo de instalação a fim de adequar o produto ao ambiente e B é o número

total de passos necessários para a correta instalação da unidade de software.

4.2.7 Métrica Externa de Coexistência

- *Coexistência apresentada*: esta métrica mede a frequência de restrições de uso ou erros encontrados pelo usuário da unidade de software enquanto esta se encontra em uso e está compartilhando recursos com outros produtos. Ela é definida pela fórmula $X = A/T$, onde X é a medida da coexistência apresentada, A é o número de restrições ou erros encontrados e T é o tempo de uso da unidade de software enquanto executa e compartilha recursos com outros produtos.

4.2.8 Métrica Externa de Capacidade para Substituir

- *Facilidade de migração*: esta métrica visa medir o quão fácil será para o usuário migrar para a unidade de software em questão. Ela é definida pela fórmula $X = A/B$, onde X é a medida da facilidade de migração, A é o número de funcionalidades unidade de software que produziram resultados satisfatórios em relação ao produto antigo e B é o número de funcionalidades testadas.

4.3 Estabelecer Níveis de Pontuação para as Métricas

Cada valor medido a partir das métricas estabelecidas deve ser mapeado numa escala, pois um valor por si só não mostra o nível de satisfação do requisito considerado [8].

Assim, cada uma das métricas definidas deve possuir uma escala individual que traduza o valor obtido em um resultado inteligível para o

indivíduo (ou indivíduos) receptor da informação. A escala utilizada será dividida em duas categorias: satisfatória e insatisfatória, ou seja, os valores calculados de cada uma das métricas serão classificados dentro de uma dessas categorias.

As tabelas 4.1 e 4.2 apresentam, respectivamente, as escalas definidas para cada uma das métricas internas e externas definidas.

Métrica	Valores insatisfatórios	Valores satisfatórios
Adaptabilidade ao ambiente de hardware	Entre 0 e 0,7	Acima de 0,7
Adaptabilidade ao ambiente de software	Entre 0 e 0,7	Acima de 0,7
Grau de portabilidade	Entre 0 e 0,6	Acima de 0,6
Esforço para instalar	Acima de 0,4	Entre 0 e 0,4
Flexibilidade de instalação	Acima de 0,4 ou abaixo de 0,2	Entre 0,2 e 0,4
Disponibilidade de coexistência	Entre 0 e 0,7	Acima de 0,7
Consistência de funcionalidades	Entre 0 e 0,9	Acima de 0,9

Tabela 4.1 – Escala para métricas internas.

Métrica	Valores insatisfatórios	Valores satisfatórios
Adaptabilidade ao ambiente de hardware	Entre 0 e 0,8	Acima de 0,8
Adaptabilidade ao ambiente de software	Entre 0 e 0,8	Acima de 0,8
Facilidade de instalação	Acima de 0,4	Entre 0 e 0,4
Coexistência apresentada	Acima de 0,3	Entre 0 e 0,3
Facilidade de migração	Entre 0 e 0,9	Acima de 0,9

Tabela 4.2 – Escala para métricas externas.

Os valores considerados foram propostos a partir de uma análise empírica, e devem ser ajustados à medida que este processo de avaliação seja aplicado em diversos produtos de software.

4.4 Obter e Julgar os Valores Obtidos

Após definir as métricas, os níveis de pontuação e os critérios para julgamento dos atributos considerados na avaliação da portabilidade de unidades de software, restam os processos de obtenção e julgamento dos valores obtidos. Esses processos foram destrinchados nas seguintes atividades:

1. Identificar as informações disponíveis sobre a unidade de software.
2. Identificar as dependências do software com o ambiente.
3. Selecionar o método de avaliação da portabilidade adequado a partir das informações disponíveis.
4. Especificar novos ambientes para a unidade de software dependendo do método de avaliação selecionado.
5. Executar a avaliação de acordo com o método de avaliação da portabilidade selecionado.
6. Sintetizar os resultados.

Neste estudo foi utilizado o *Software Process Engineering Metamodel* [10] para a descrição do processo de avaliação da portabilidade de unidades de software. O SPEM é um metamodelo utilizado para definir processos, seus componentes e as interações entre eles. Este modelo utiliza estereótipos para representar atividades, papéis, ferramentas, documentos.

No modelo do processo de avaliação da portabilidade de unidades de software descrito neste trabalho podem ser encontrados os seguintes

estereótipos: bonequinhos que representam os papéis dos indivíduos envolvidos no processo; polígonos semelhantes a setas que representam atividades desempenhadas pelos papéis; páginas que representam os artefatos gerados a partir da execução das atividades.

A figura 4.1 apresenta o processo de avaliação da portabilidade de unidades de software descrito a partir do SPEM.

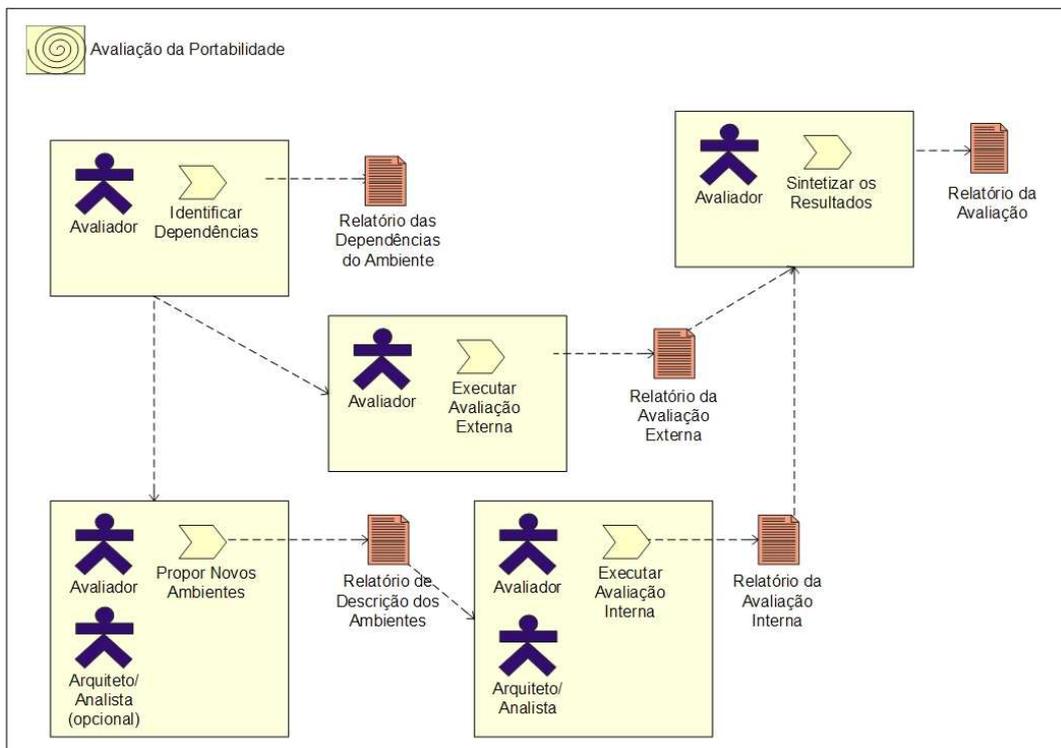


Figura 4.1 – Processo de Avaliação da Portabilidade.

Como forma de apoio ao processo de avaliação, foi criada uma *checklist* de acompanhamento, encontrada no apêndice A. São apresentadas as atividades do processo e seus passos internos e, à medida que a avaliação prossegue, o avaliador registra seu progresso na *checklist*.

4.4.1 Identificar as Informações Disponíveis

Verificar junto ao requisitante da avaliação a disponibilidade da documentação necessária para a avaliação da portabilidade: especificação

de requisitos de software, documento de arquitetura, programa-fonte, manuais de sistema.

Esses documentos são importantes porque oferecem informações técnicas relevantes que serão utilizadas durante o processo de avaliação da portabilidade de unidades de software definido neste documento.

4.4.2 Identificar Dependências

Identificar as funcionalidades do software que apresentam alguma dependência com o ambiente onde a unidade de software estará inserida, inicialmente descrito na especificação de requisitos ou nos manuais do sistema. Essas dependências podem existir no nível de hardware ou software.

Neste momento, a presença de um arquiteto e/ou analista, cedido pelo requisitante da avaliação, é necessária nos casos onde as informações obtidas a partir da documentação são incompletas ou inexistentes. Para coletar essas informações, há a aplicação de um questionário que deve ser preenchido a partir das respostas do arquiteto e/ou analista. Este questionário é apresentado no apêndice B.

A partir das informações coletadas, provenientes da documentação disponível e/ou questionário, deve ser realizado o mapeamento das funcionalidades que apresentaram alguma dependência com os respectivos componentes onde elas (as funcionalidades) estão inseridas.

Ao final deste processo, um relatório contendo uma descrição detalhada das dependências da unidade de software com o ambiente deve ser produzido. Um modelo para este relatório pode ser encontrado no apêndice C.

A tabela 4.3 apresenta um resumo desta atividade.

Atividade: Identificar dependências	
Objetivos:	
<ul style="list-style-type: none"> • Identificar as funcionalidades do sistema que possuem alguma dependência com o ambiente inicialmente descrito. 	
Entradas:	Saídas:
<ul style="list-style-type: none"> • Especificação de requisitos de software • Documento de arquitetura • Programa-fonte (opcional) 	<ul style="list-style-type: none"> • Relatório de dependências • Questionário sobre portabilidade • Checklist do processo
Passos:	
<ul style="list-style-type: none"> • Executar o questionário sobre portabilidade (quando necessário) • Levantar as funcionalidades do sistema que dependem do ambiente • Mapear as funcionalidades dependentes nos componentes do sistema 	

Tabela 4.3 – Resumo da atividade Identificar dependências.

4.4.3 Selecionar o Método de Avaliação

Deve ser discutido com o requisitante da avaliação, após identificar e documentar as dependências da unidade de software com o ambiente, o método de avaliação da portabilidade que será utilizado. Este estudo propõe três métodos: avaliação externa da portabilidade, avaliação interna da portabilidade e avaliação mista da portabilidade.

O método de **avaliação externa** da portabilidade consiste em analisar a portabilidade do sistema a partir de métricas externas definidas na seção 4.2 deste documento, ou seja, sob o ponto de vista do usuário da unidade de software.

O método de **avaliação interna** da portabilidade visa analisar a portabilidade da unidade de software utilizando-se das métricas internas também definidas na seção 4.2 deste estudo, ou seja, a partir de atributos intrínsecos do software. Para a aplicação deste método, faz-se necessário o acesso ao programa-fonte da unidade de software.

O terceiro método disponível, chamado método de *avaliação mista* da portabilidade, consiste na aplicação em paralelo dos dois métodos anteriormente descritos. Este método é mais abrangente e, por conseguinte, oferece mais informações para o requisitante da avaliação, porém requer um esforço maior por parte do avaliador (tempo) e do requisitante da avaliação (custo).

A tabela 4.4 apresenta um resumo desta atividade.

Atividade: Selecionar o método de avaliação	
Objetivos:	
<ul style="list-style-type: none"> • Discutir com o requisitante o método de avaliação que será utilizado. 	
Entradas:	Saídas:
<ul style="list-style-type: none"> • Relatório de dependências 	<ul style="list-style-type: none"> • Checklist do processo
Passos:	
<ul style="list-style-type: none"> • Descrever os métodos disponíveis para o requisitante • Identificar junto ao requisitante qual(is) métodos utilizar 	

Tabela 4.4 – Resumo da atividade Selecionar o método de avaliação.

Todos esses métodos de avaliação devem levar em consideração as dependências entre a unidade de software e o ambiente levantadas nos passos anteriores, ou seja, considerar apenas os módulos, componentes ou camadas que foram identificados.

4.4.4 Propor Novos Ambientes

Quando o requisitante da avaliação opta por avaliar a unidade de software através dos métodos de avaliação interna ou avaliação mista, faz-se necessário criar novos ambientes de hardware e software que ofereçam mudanças em relação ao ambiente originalmente descrito para unidade de software em avaliação.

É importante que a especificação de cada novo ambiente seja explícita e bem definida, a fim de evitar ambigüidades e ambientes que

apresentem mudanças em demasia, tornando a avaliação complexa e custosa.

Para atender esta sugestão, foi proposto um relatório simples que apresenta de forma resumida as características do novo ambiente. Este relatório pode ser encontrado no Apêndice D.

A tabela 4.5 a seguir apresenta um resumo desta atividade.

Atividade: Propor novos ambientes	
Objetivos: <ul style="list-style-type: none">• Criar novos ambientes de hardware e software que oferecem mudanças em relação ao ambiente originalmente descrito.	
Entradas: <ul style="list-style-type: none">• Relatório de dependências	Saídas: <ul style="list-style-type: none">• Relatório de descrição dos ambientes• Checklist do processo
Passos: <ul style="list-style-type: none">• Verificar as dependências e propor ambientes simples	

Tabela 4.5 – Resumo da atividade Propor novos ambientes.

4.4.5 Executar a Avaliação

Após definir qual método de avaliação será utilizado, o avaliador deve aplicar as métricas definidas para ele (o método) e coletar os valores dos atributos correspondentes.

Esses valores devem ser analisados e classificados dentro da escala especificada na seção 4.3 (ver tabelas 4.1 e 4.2) e o relatório parcial correspondente deve ser confeccionado. Mais uma vez, o modelo do relatório não será apresentado aqui, pois se trata de um documento interno do processo de avaliação do LAPS e possui direitos autorais que devem ser respeitados.

A tabela 4.6, abaixo, apresenta um resumo desta atividade.

Atividade: Executar a avaliação	
Objetivos:	
<ul style="list-style-type: none"> • Aplicar o método de avaliação definido e coletar os valores das métricas associadas. 	
Entradas:	Saídas:
<ul style="list-style-type: none"> • Relatório de dependências • Relatório de descrição dos ambientes 	<ul style="list-style-type: none"> • Relatório da avaliação (interna e/ou externa) • Checklist do processo
Passos:	
<ul style="list-style-type: none"> • Verificar a avaliação selecionada • Aplicar as métricas definidas • Coletar os valores das métricas • Analisar e classificar os valores obtidos 	

Tabela 4.6 – Resumo da atividade Executar a avaliação.

4.4.6 Sintetizar os Resultados

Ao concluir a avaliação da portabilidade da unidade de software considerada a partir do método de avaliação definido, os resultados obtidos devem ser compilados em um único documento e disponibilizados ao requisitante da avaliação. Esses resultados devem ser apresentados de forma clara e objetiva e justificados no relatório final da avaliação, o qual deve trazer como anexo o(s) relatório(s) da(s) avaliação(s) executada(s). O modelo do relatório final da avaliação não será apresentado por se tratar de um documento interno do processo de avaliação do LAPS.

A tabela 4.7 apresenta esta atividade de forma resumida.

Atividade: Sintetizar os resultados	
Objetivos:	
<ul style="list-style-type: none"> • Apresentar ao requisitante da avaliação o resultado final desta, de forma clara e objetiva. 	
Entradas:	Saídas:
<ul style="list-style-type: none"> • Relatório(s) da(s) avaliação(s) 	<ul style="list-style-type: none"> • Relatório final da avaliação • Checklist do processo
Passos:	
<ul style="list-style-type: none"> • Compilar os resultados • Disponibilizar o relatório final para o requisitante da avaliação 	

Tabela 4.7 – Resumo da atividade Sintetizar os resultados.

4.5 Considerações Finais

Este capítulo apresentou o objetivo maior deste estudo que foi o processo de avaliação da portabilidade de unidades de software baseado nas normas internacionais ISO/IEC 9126 e ISO/IEC 14598.

Foram definidas métricas internas e externas, assim como métodos de avaliação que fazem uso dessas métricas. Foi proposta uma escala para classificar o resultado da medição das métricas definidas a fim de tornar o resultado final mais inteligível.

Documentos de apoio e relatórios foram definidos como forma de controle, divulgação e armazenamento das informações obtidas nas atividades intermediárias do processo de avaliação da portabilidade de unidades de software.

O próximo capítulo apresentará um estudo de caso onde se pretendeu aplicar o passo a passo definido pelo processo aqui especificado, com a finalidade de validar as métricas propostas e o próprio processo de avaliação da portabilidade.

5 Estudo de Caso

A fim de validar o processo de avaliação da portabilidade de unidades de software definido neste documento, este foi aplicado a um produto de software real o qual foi gentilmente cedido pela empresa de desenvolvimento de sistemas Apply Solutions.

O software em questão é um sistema de informação inserido dentro do contexto de um portal para a Web. Este portal oferece serviços a seus usuários que vão desde cadastro, recuperação e manutenção de dados até serviço de anúncios de compra e venda e emissão de boletos de pagamento. O nome do produto foi mantido em sigilo a pedido da empresa, assim como detalhes mais específicos de seu funcionamento e implementação.

5.1 Escopo da Avaliação

Durante a conversa com a pessoa responsável pela coordenação técnica projeto de desenvolvimento do produto avaliado, ficou evidenciado que o produto encontra-se em estágio de conclusão e que ainda pode sofrer modificações.

Após explicar a metodologia de avaliação de forma resumida, as métricas internas e externas foram apresentadas. Imediatamente, o coordenador técnico identificou que a avaliação interna seria mais interessante para o produto, pois poderia sugerir pontos de melhorias no sistema.

Esta opção foi considerada, embora trouxesse limitações para a validação do processo de avaliação da portabilidade de unidades de

software descrito neste documento, pois permitiria que apenas uma das avaliações fosse executada e, conseqüentemente, validada. Contudo, devido à falta de tempo para buscar outra empresa interessada em avaliar a portabilidade de algum de seus produtos, esta avaliação foi realizada dentro das limitações encontradas.

5.2 A Avaliação

Após fechar alguns pontos junto à empresa quanto ao caráter sigiloso das informações disponibilizadas sobre o produto, deu-se início ao processo de avaliação da portabilidade.

O primeiro passo foi levantar a documentação disponível sobre o produto. De imediato foi conhecida a inexistência de um documento de requisitos e de um documento da arquitetura do sistema. Também foi exposta a ausência de manuais do sistema. Os documentos disponibilizados foram: um documento de casos de uso e o programa-fonte.

Imediatamente foi identificada a necessidade de entrevistar o arquiteto ou um analista do sistema a fim de levantar as informações necessárias para a continuidade da avaliação. Essas informações foram obtidas a partir da aplicação do questionário sobre a portabilidade do sistema.

O responsável por responder o questionário foi o arquiteto do sistema, o qual proferiu, de imediato, críticas ao questionário. Basicamente as críticas foram relacionadas à ambigüidade das questões. Apesar das críticas, a realização prosseguiu e os documentos de apoio começaram a ser preenchidos.

As dependências da unidade de software com o ambiente foram identificadas e documentadas e um novo ambiente, baseado nessas dependências, foi especificado.

A partir deste momento, iniciou-se a aplicação das métricas definidas a respeito da avaliação interna. Com a ajuda do arquiteto da aplicação, rapidamente foram levantados os parâmetros necessários para a realização dos cálculos propostos nestas métricas.

5.3 Resultados

A seguir serão apresentados os valores obtidos a partir das métricas definidas, como também sua adequação à escala definida para cada uma das métricas.

A tabela 5.1 apresenta esses resultados.

Métrica	Valor obtido	Classificação
Adaptabilidade ao ambiente de hardware	1	Satisfatório
Adaptabilidade ao ambiente de software	0,78	Satisfatório
Grau de portabilidade	0,625	Satisfatório
Esforço para instalar	0	Insatisfatório
Flexibilidade de instalação	0,33	Satisfatório
Nível de coexistência	---	Não se aplica
Consistência de funcionalidades	---	Não se aplica

Tabela 5.1 - Resultados da aplicação das métricas.

A métrica “Adaptabilidade ao ambiente de hardware” recebeu valor 1 porque o produto foi desenvolvido utilizando-se a linguagem Java, a qual é interpretada e executa sobre uma máquina virtual, a qual abstrai da aplicação os detalhes do hardware.

As métricas “Nível de coexistência” e “Consistência de funcionalidades” não foram consideradas, pois o produto de software em questão será instalado numa máquina dedicada a ele e, por ser um

produto novo e sob encomenda para um cliente específico da empresa, não irá substituir qualquer outro produto de software existente.

O resultado final da avaliação, ou seja, a nota obtida pelo produto não será demonstrada porque o processo de cálculo da nota final faz parte do processo do LAPS e não pode ser divulgado. Contudo é possível obter uma idéia do resultado final a partir dos resultados intermediários produzidos pela aplicação das métricas definidas.

Apesar de não terem sido validadas todas as métricas definidas no processo de avaliação da portabilidade, este estudo de caso foi bastante útil para que pudessem ser identificados pontos de melhoria, principalmente no que diz respeito aos documentos de apoio ao processo.

O fato de nem todas as métricas terem sido aplicadas, não é tão impactante para a aceitação deste processo de avaliação, visto que as métricas foram retiradas de sugestões nas normas ISO/IEC 9126-2 e ISO/IEC 9126-3.

Os documentos intermediários produzidos no decorrer desta avaliação não serão apresentados a pedido da empresa, por questões comerciais inerentes ao contrato da Apply Solutions com o comprador do sistema.

5.4 Considerações Finais

Este capítulo apresentou um estudo de caso onde se objetivou a aplicação do processo de avaliação da portabilidade de unidades de software, definido neste documento, em um software real para que fosse possível validar as métricas utilizadas.

Devido ao pedido da empresa que disponibilizou o produto, apenas a avaliação interna foi aplicada, o que veio a impedir a aplicação e validação de todas as métricas definidas.

O próximo capítulo apresentará as conclusões deste trabalho, indicando as principais contribuições deste estudo, as dificuldades encontradas e os possíveis trabalhos futuros.

6 Conclusão

Visando atender aos novos requisitos do mercado atual de tecnologia da informação, que vem crescendo e buscando cada vez mais oferecer produtos com altos índices de qualidade, este trabalho apresentou uma metodologia de avaliação da portabilidade de unidades de software baseada nas normas de qualidade e avaliação de software da *International Organization for Standardization* (ISO).

A proposta de uma metodologia de avaliação de produtos de software apoiou-se no fato de que uma das ações mais efetivas que vêm sendo executadas pelas empresas de TI é a implantação de processos de garantia da qualidade de software visando aumentar a competitividade de seus produtos.

A característica portabilidade foi considerada neste estudo porque é incontestável que unidades de software nunca executam sozinhas, mas sempre inseridas em um ambiente onde compartilham recursos com outros softwares, como também estudos mostraram, que unidades de software que apresentam níveis satisfatórios de portabilidade, possuem maior vida útil.

Todo o processo de avaliação da portabilidade de unidades de software apresentado neste estudo foi especificado tomando-se os devidos cuidados para que estivesse em conformidade com as normas ISO/IEC 9126 e ISO/IEC 14598.

6.1 Principais Contribuições

A principal contribuição deste trabalho foi a própria definição do processo de avaliação da portabilidade de unidades de software. Este processo apresenta métricas internas e externas para a avaliação da portabilidade baseadas nas normas ISO/IEC 9126-2 e ISO/IEC 9126-3, como também descreve como coletar e interpretar os valores obtidos através da aplicação dessas métricas como sugerido pela ISO/IEC 14598-5. Este processo foi absorvido pelo Laboratório de Avaliação de Produtos de Software, o LAPS, como sendo um dos módulos de avaliação de produtos de software que são oferecidos pelo laboratório, o que vem a ser um grande indício do potencial deste estudo.

Este estudo também abriu precedentes para a realização de outros estudos sobre portabilidade, dentro do contexto da qualidade de software, tanto ao nível de produto quanto ao nível de processo, pois oferece diversos pontos de discussão entre pesquisadores interessados na característica de qualidade de software portabilidade.

6.2 Dificuldades Encontradas

Durante o desenvolvimento do processo descrito neste documento, várias dificuldades foram encontradas. Em sua maior parte foram superadas, essencialmente as que ofereciam maiores impactos à definição do processo de avaliação da portabilidade. Podem ser citadas como dificuldades superadas: a escassez de literatura relevante, disponível sobre avaliação da portabilidade, o custo de obtenção das normas utilizadas neste estudo e o pouco tempo para definir o processo de avaliação por conta das necessidades do LAPS.

Contudo, existem dificuldades que não puderam ser superadas. Basicamente, essas dificuldades estão associadas à validação do processo definido neste estudo, mais especificamente no ajuste das escalas utilizadas para classificar os resultados das medições a partir das métricas definidas devido ao fato de apenas um único produto de software ter sido avaliado a partir da metodologia criada e também por apenas uma das avaliações, a avaliação interna, ter sido aplicada.

6.3 Trabalhos Futuros

Como mencionado, este trabalho abre precedentes para a realização de estudos sobre portabilidade porque oferece diversos pontos de discussão, como, por exemplo, o refinamento das métricas e escalas utilizadas durante o processo de avaliação definido.

Há a possibilidade de introdução de novas métricas, da criação de uma escala mais expressiva, do refinamento dos documentos de apoio ao processo e do aprimoramento do próprio processo, pois este é uma proposta e está longe de ser o processo de avaliação da portabilidade definitivo.

Existe a necessidade de se aplicar o processo a um número maior de produtos de software a fim de torná-lo mais robusto e de acordo com as expectativas dos requisitantes da avaliação, como também de aplicar todas as avaliações oferecidas (interna, externa e mista) para que os resultados obtidos possam ser analisados e comparados.

Também pode ser explorada a possibilidade de introdução deste processo de avaliação durante todo o ciclo de vida de unidades de

software, a fim de se obter unidades de software com altos índices de portabilidade e, conseqüentemente, vida útil.

Referências

- [1] BARTIÉ, A. (2002), *Garantia da Qualidade de Software*, Editora Campus.
- [2] KOSCIANSKI, A., VILAS-BOAS, A., RÊGO, C. M., ASANOME, C., SCALET, D., ROMERO, D., CIESLAK, J. M., PALUDO, M., FROSSARD, R. S., VOSTOUPAL, T. M. (1999), *Guia para Utilização das Normas sobre Avaliação de Qualidade de Produto de Software - ISO/IEC 9126 e ISO/IEC 14598*, Associação Brasileira de Normas Técnicas (ABNT).
- [3] ISO/IEC (2001), *NBR ISO/IEC 14598-5, Tecnologia de informação - Avaliação de produto de software, Parte 5: Processo para avaliadores*, Associação Brasileira de Normas Técnicas (ABNT).
- [4] MOONEY, J. D. (2001), *Software Portability Home Page* - <http://www.cs.wvu.edu/~jdm/research/portability/home.html>, último acesso em 05/03/2005.
- [5] MOONEY, J. D. (1993), *Issues in the Specification and Measurement of Software Portability*, West Virginia University.
- [6] SAMPAIO, V. V., MOURA, H. P. (2004), *Um Modelo Estruturado de Serviços para Avaliação de Produtos de Software*, Centro de Informática - Universidade Federal de Pernambuco.
- [7] ISO/IEC (2003), *NBR ISO/IEC 9126-1, Engenharia de Software - Qualidade de produto, Parte 1: Modelo de qualidade*, Associação Brasileira de Normas Técnicas (ABNT).

- [8] ISO/IEC (2001), *NBR ISO/IEC 14598-1, Tecnologia de informação - Avaliação de produto de software, Parte 1: Visão geral*, Associação Brasileira de Normas Técnicas (ABNT).
- [9] MOONEY, J. D. (1990), *Strategies for Supporting Application Portability*, IEEE Computer, Volume 23, No. 11, Nov. 1990.
- [10] Object Management Group (2005), *Software Process Engineering Metamodel Specification*, Object Management Group.
- [11] ISO/IEC (2003), *NBR ISO/IEC 9126-3, Engenharia de Software - Qualidade de produto, Parte 3: Métricas internas*, Associação Brasileira de Normas Técnicas (ABNT).
- [12] ISO/IEC (2003), *NBR ISO/IEC 9126-2, Engenharia de Software - Qualidade de produto, Parte 2: Métricas externas*, Associação Brasileira de Normas Técnicas (ABNT).

Apêndice A

Checklist de portabilidade

Neste apêndice é apresentada a checklist de portabilidade definida para apoiar o acompanhamento do processo de avaliação da portabilidade de unidades de software definido neste estudo.

Apêndice A – Checklist do processo de portabilidade

Documentação disponível

- Especificação de Requisitos de Software
- Documento de Arquitetura
- Programa-Fonte
- Manuais de sistema

Identificar dependências

- Disponibilidade do arquiteto e/ou analista
- Realizar questionário
- Produzir relatório de dependências

Selecionar o método de avaliação

- Avaliação externa
- Avaliação interna
- Avaliação mista

Executar a avaliação

Interna

- Propor novos ambientes
- Adaptabilidade ao ambiente de hardware
- Adaptabilidade ao ambiente de software
- Grau de portabilidade
- Esforço para instalar
- Flexibilidade de instalação
- Nível de coexistência
- Consistência de funcionalidades

Externa

- Adaptabilidade ao ambiente de hardware
- Adaptabilidade ao ambiente de software
- Facilidade de instalação
- Nível de coexistência
- Facilidade de migração

Sintetizar os resultados

- Analisar o(s) relatório(s) da(s) avaliação(s)
- Produzir o relatório final da avaliação

Apêndice B

Questionário sobre portabilidade

Neste apêndice é apresentado o questionário sobre portabilidade definido para apoiar o processo de avaliação da portabilidade de unidades de software definido neste estudo.

Este questionário foi desenvolvido para os casos em que o requisitante da avaliação não possui toda a documentação recomendada sobre a unidade de software, visando identificar os pontos necessários para a avaliação da portabilidade.

Apêndice C

Relatório de dependências

Neste apêndice é apresentado o relatório de dependências da unidade de software com o ambiente definido para apoiar o processo de avaliação da portabilidade de unidades de software definido neste estudo.

A seguir é apresentado o modelo de relatório de dependências que deve ser utilizado durante o processo de avaliação da portabilidade de unidades de software.

Apêndice D

Relatório de descrição dos ambientes

Neste apêndice é apresentado o relatório de descrição dos ambientes definido para apoiar o processo de avaliação da portabilidade de unidades de software definido neste estudo.

Este documento tem por objetivo apresentar a descrição de um ambiente proposto para a avaliação da portabilidade de unidades de software, e será utilizado durante a aplicação do método de avaliação interna.

Apêndice D – Relatório de descrição dos ambientes

Ambiente

Identificador do ambiente:
Tipo de dependência considerada <input type="checkbox"/> Sistema de Arquivos <input type="checkbox"/> Sistema Operacional <input type="checkbox"/> SGBD <input type="checkbox"/> Hardware Específico <input type="checkbox"/> Rede <input type="checkbox"/> Outros Softwares
Módulos, componentes ou camadas relacionados
Modificações propostas

Data e assinaturas

Alegamos estar cientes do conteúdo deste trabalho e do propósito ao qual ele está destinado.

Marcílio José Albuquerque Gomes Filho
Graduando

Prof. Dr. Alexandre Marcos Lins de Vasconcelos
Orientador

Recife, 11 de março de 2005