

Processo de Avaliação e Seleção de Componentes de Software Adquiridos de Terceiros

Thiago Massao Hirata¹, Selma Shin Shimizu Melnikoff¹

¹ Escola Politécnica da Universidade de São Paulo (Poli - USP)
Depto. de Engenharia de Computação e Sistemas Digitais (PCS)
Laboratório de Tecnologia de Software (LTS)
Av. Prof. Luciano Gualberto, 158 – CEP 05508-900 – São Paulo – SP – Brasil
{thiago.hirata, selma.melnikoff}@poli.usp.br

Abstract. *This paper presents an evaluation process for software components. The objective of an evaluation process is to provide base for decisions involving the selection and acquisition of software components in projects using the component based development. One of the main characteristics of the proposed process is the adaptability for different contexts, supporting projects with low budget, or with high risks as well. To illustrate its use, this paper presents the results of the evaluation of commercial and open source components.*

Resumo. *Este artigo apresenta um processo para a execução de avaliações de componentes de software. O objetivo de um processo de avaliação é prover base para decisões de seleção e uso de componentes em projetos que utilizam o desenvolvimento baseado em componentes. Uma das características principais do processo proposto é a adaptabilidade para diferentes contextos, suportando projetos com restrições de custo e prazo, assim como projetos envolvendo altos riscos. Para ilustrar seu uso, este artigo apresenta os resultados da avaliação de componentes comerciais e componentes de código aberto.*

1. Introdução

O desenvolvimento de sistemas baseado em componentes (CBD – *Component Based Development*) é uma abordagem que tem como objetivo diminuir os custos e tempo de desenvolvimento e aumentar a qualidade dos sistemas de software. Sua premissa é a criação de novos sistemas a partir de componentes de software reutilizáveis, sejam eles desenvolvidos pela própria organização, ou adquiridos de fornecedores externos.

No entanto, a decisão de utilizar um componente de software desenvolvido por terceiros não pode ser feita sem uma avaliação adequada. A avaliação é importante para detectar e evitar componentes de baixa qualidade, que não atendam aos requisitos do software em desenvolvimento. Sem esse cuidado, os problemas serão detectados em fases já avançadas do projeto, resultando em custos e atrasos inesperados.

Nesta área de estudos, existem trabalhos sobre a avaliação de componentes de software baseados na utilização de modelos de qualidade. Entre eles, existem propostas de modelos de qualidade para componentes COTS [Alvaro 2005], [Meyer 2003],

[Torchiano 2002], metodologias para a criação e a avaliação de modelos de qualidade para componentes [Franch 2003], [Torchiano 2003].

No entanto, não fica claro nesses trabalhos como implantar na prática a avaliação de componentes dentro dos processos de uma organização.

O objetivo deste trabalho é apresentar uma proposta de processo para a avaliação de componentes de software adquiridos de terceiros, baseado em modelos de qualidade. Este processo não adota um modelo de qualidade específico, e sim considera que a definição e a escolha do modelo devem atender as necessidades de cada contexto.

Este artigo está dividido da seguinte forma: na seção 2 é feito um breve sumário relacionado à pesquisa sobre modelos de qualidade de componentes específicos para componentes de software. A proposta de processo de avaliação de componentes é apresentada na seção 3. A seção 4 apresenta resultados de experimentos da utilização do processo proposto. E, finalmente, a seção 5 apresenta as considerações finais, e trabalhos futuros na área.

2. Modelos de qualidade de componentes de software

A proposta de processo de avaliação deste trabalho é orientada a produto, ou seja, na análise direta das características de um produto. Esse tipo de avaliação é diferente de metodologias que avaliam os processos produtivos de uma organização, como é o caso do CMMI e SPICE, que se baseiam na premissa de que a qualidade de um processo tem como consequência a geração de produtos de qualidade.

A avaliação orientada a produto utiliza como base modelos de qualidade. Um modelo de qualidade define quais fatores de qualidade serão analisados durante a avaliação.

Alguns fatores de qualidade são abstratos, ou seja, não podem ser observados diretamente. Esses fatores, no entanto, podem ser decompostos em fatores concretos, que são mensuráveis, denominados atributos. Por exemplo, o número de exemplos de uso disponíveis na documentação é um atributo que tem influência sobre a usabilidade, um fator de qualidade abstrato.

A norma ISO/IEC 9126 estabelece um modelo de qualidade genérico para produtos de software. Este modelo define seis fatores de qualidade principais, denominados características: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. Cada característica, por sua vez, é dividida em conjuntos mais granulares de fatores de qualidade abstratos, chamados de sub-características.

A norma, no entanto, não define os atributos de qualidade de cada sub-característica. Para a sua utilização na prática, o modelo deve ser especializado de acordo com as particularidades de cada organização e contexto de uso, definindo-se um conjunto de atributos, e as métricas utilizadas para a sua medição.

O modelo de qualidade da norma, na forma em que é proposto, é de difícil aplicação para o caso de componentes, necessitando de algumas modificações [Alvaro 2005a] [Torchiano 2002] [Bertoa 2002]. Um modelo de qualidade para componentes deve levar em consideração as particularidades para esse tipo de produto. Por exemplo,

o modelo proposto por Alvaro (2005), que é uma adaptação do modelo da norma ISO/IEC 9126, acrescenta a capacidade de reuso do componente e a dependência em relação a outros componentes, fatores importantes para esse tipo de produto, entre outros. Bertoa (2002) também contribui com seu modelo de qualidade baseado na norma ISO/IEC 9126, propondo um conjunto de atributos e suas correspondentes métricas para a avaliação de componentes COTS.

Apesar da ISO/IEC 9126 ser uma referência importante, ela não é o único ponto de partida para a geração de modelos de qualidade. Meyer (2003), por exemplo, descreve uma versão preliminar de um modelo de qualidade original, para ser utilizado na certificação de componentes comerciais e de código aberto disponíveis no mercado. Já Torchiano (2002), levantou com estudantes um conjunto de atributos importantes para a caracterização de componentes COTS.

2.1 Níveis de especialização de um modelo de qualidade

Um modelo de qualidade especializado para componentes de software pode sofrer novas especializações, com o objetivo de atender os objetivos particulares de cada situação, ou conjunto de situações. De acordo com seu objetivo, modelos podem diferir no nível de rigor, ou então incluir atributos e fatores de qualidade que sejam particularmente significativos para um determinado domínio ou tecnologia.

Botella (2002) classifica modelos de qualidade em diferentes tipos: livre de contexto, específico a uma categoria de produtos (ferramentas para comunicação, compiladores), específico para um domínio (servidores de e-mail, bibliotecas para aplicações matemáticas), específico a um tipo de organização (empresas grandes ou pequenas) e modelos finais (específico para uma avaliação particular).

Essa divisão representa um possível método para a criação de modelos concretos, ou seja, modelos que tenham todos os seus atributos definidos. Neste caso, conforme o nível de especialização aumenta, é maior a granularidade dos fatores de qualidade. No final da cadeia, o modelo final é aquele que possui os atributos utilizados durante uma avaliação.

Um modelo pode ser especializado para uma tecnologia específica, definindo atributos mais adequados. Por exemplo, Washizaki (2003) apresenta um modelo de qualidade específico para a arquitetura de componentes JavaBeans, contendo atributos relacionados à reusabilidade. Os atributos propostos levam em consideração as boas práticas pregadas para o desenvolvimento deste tipo de componente, e são facilmente medidos quando se usa essa tecnologia.

2.2 Atributos de qualidade

Os atributos são os elementos da base da hierarquia do modelo de qualidade. Cada fator de qualidade é avaliado a partir de um conjunto de atributos mensuráveis, seguindo as restrições e necessidades para os quais o modelo é elaborado. Por exemplo, Alvaro (2005) propõe avaliar a segurança de um componente verificando a presença de algoritmos de criptografia, a proporção de interfaces que possuem controle de acesso e a presença de mecanismos de gravar logs de acesso e uso do sistema e seus dados.

Franch (2003) identifica dois tipos de atributos: básicos e derivados. Os básicos são aqueles que podem ser diretamente medidos, como contar o número de páginas do manual de instruções. Já os derivados têm um nível maior de abstração, mas ainda possuem uma métrica associada, cujo valor é determinado a partir dos valores de atributos básicos. O autor usa como exemplo o nível de recursos administrativos de um servidor de e-mail, cuja leitura é feita a partir da verificação da presença dos seguintes recursos: tempo máximo de armazenamento de mensagens, tempo máximo de vida de contas inativas, quotas de caixas de mensagens, tamanho máximo de mensagens e gerenciamento de grupos de servidores.

Os atributos também podem ser divididos entre os observáveis em *runtime*, lidos em tempo de execução, e os observáveis durante o ciclo de vida do produto, lidos a partir dos artefatos gerados durante o desenvolvimento do produto [Alvaro 2005].

2.3 Métricas

Alvaro (2005) utiliza em seu modelo três tipos de métrica:

- **Presença:** Métrica utilizada para atributos que podem estar presentes ou não no produto, recebendo um valor booleano. Por exemplo, a presença de algoritmo de criptografia. Em caso positivo, também deve ser associado um texto a este atributo, descrevendo detalhes sobre a implementação do elemento no componente;
- **Valor:** Utilizada quando o atributo pode ser representado diretamente por um valor numérico, com uma unidade (exemplo, número de páginas, tamanho em bytes, segundos);
- **Taxa:** Utilizada para porcentagens.

Existem modelos na literatura que utilizam atributos cuja métrica não se encontra dentro destes três casos. Por esse motivo, foram adicionados mais três tipos de métrica:

- **Codificação:** Uma escala customizada para atributos derivados. Por exemplo, a medida do atributo abstrato “capacidade de configuração” pode ser considerada “insatisfatória”, “satisfatória” ou “ótima”, dependendo dos valores de outros atributos básicos. Meyer (2003) tem em seu modelo atributos de alto nível de abstração, como “comportamento”, que recebem uma pontuação de 1 a 6 de acordo com a verificação de atributos básicos, como “exemplos na documentação” e “especificação das pré-condições para o uso”;
- **Nominal:** Usado quando o atributo não pode ser representado por um valor numérico, apenas por um texto (exemplo, tipo de licença de uso [Torchiano, 2003]). Apesar de não proporcionar uma comparação óbvia, como nos outros tipos de métricas, atributos nominais podem ser utilizados quando merecem registro, e podem influenciar na seleção de um componente;
- **Curva:** Certos atributos podem ser representados a partir de uma série de resultados. A avaliação da escalabilidade de um produto, por exemplo,

pode ser baseada em diferentes leituras de seu tempo de resposta, variando a quantidade de dados utilizada, ou o número de processadores de um cluster.

Washizaki (2003) e Bertoa (2005), em seus trabalhos, apresentam métodos para a validação de métricas. No entanto, entre diferentes propostas de modelos de qualidade para componentes, geralmente não há validação suficiente [Goulão 2004][Goulão 2004b].

2.4 Escopo de um atributo

Enquanto alguns atributos são medidos do componente como um todo, outras leituras são dependentes de uma função específica. Outros atributos também dependem do contexto de uso. O escopo de um atributo define em qual dessas situações ele se enquadra.

O tipo de escopo pode ser:

- **Componente:** O atributo está associado ao componente. Exemplos: tamanho do binário do componente, dependências do componente, número de páginas da documentação;
- **Função:** O valor do atributo pode variar dependendo da função do componente. Para componentes com várias funções, caso de bibliotecas de controles para páginas web, esse valor deve ser medido diversas vezes. A avaliação individual de cada função permite visualizar casos em que a qualidade não seja uniforme em todo o componente. Por exemplo, a qualidade da documentação de um componente de geração de gráficos pode variar entre os vários tipos de gráficos disponíveis;
- **Contexto de uso:** O valor do atributo depende do contexto em que ele foi medido. Leituras obtidas por meio de testes, como tempo de resposta e ocupação da memória, devem ser acompanhadas da descrição das condições sob as quais foram medidas, como capacidade de processamento do computador, quantidade de dados utilizada, entre outros. Um atributo deste tipo ainda está relacionado ao componente inteiro, ou a uma função específica.

2.3 Meta-modelo de qualidade

A Figura 1 representa um meta-modelo para modelos de qualidade, no formato de um diagrama de classes em UML. Nele estão representados os principais elementos de um modelo de qualidade e os relacionamentos entre eles.

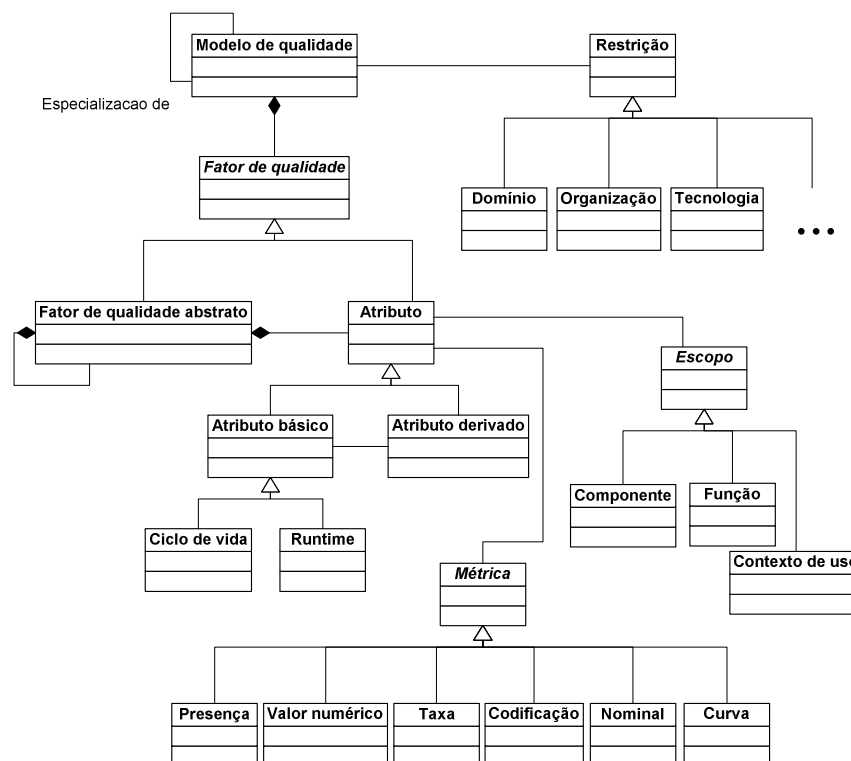


Figura 1: Meta-modelo de qualidade para componentes

3. Processo de avaliação de componentes de software

Uma vez discutido sobre modelos de qualidade para componentes de software, nesta seção serão descritas as etapas que compõem o processo de avaliação de componentes proposto.

A Figura 2 apresenta o encadeamento das fases do processo de avaliação proposto. Já a Tabela 1, ao final da seção, mostra um resumo do processo, listando as entradas e saídas de cada fase.

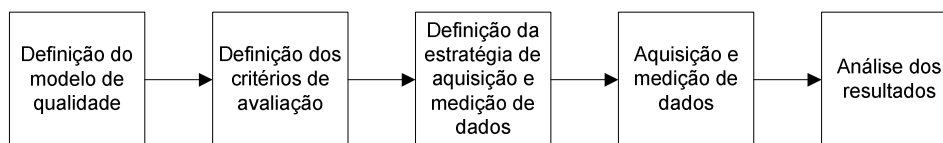


Figura 2: Processo de avaliação de componentes

3.1 Definição do modelo de qualidade

Esta etapa consiste na escolha de quais atributos de avaliação serão medidos. O contexto dirige a definição do modelo de qualidade a ser utilizado. Isso depende de fatores como as tecnologias em uso, o domínio da aplicação, as funções que o componente irá desempenhar e os requisitos particulares de um sistema em desenvolvimento. Além disso, o tempo e os recursos disponíveis para a avaliação, assim como os riscos envolvidos no projeto, determinam a quantidade de atributos do modelo de qualidade.

3.2 Definição dos critérios de avaliação

Os critérios de avaliação estabelecem como os dados levantados durante a avaliação serão utilizados durante a análise dos resultados. Em um caso em que diferentes componentes semelhantes estão sendo comparados, os critérios definem como deve ser o processo de escolha com base nos atributos lidos. Em outros casos, os critérios definem as condições mínimas que um componente precisa apresentar para ser utilizado em um projeto.

Os critérios de avaliação são obtidos a partir da interpretação dos requisitos do sistema em desenvolvimento, além da experiência obtida em avaliações anteriores dentro da organização.

3.3 Definição da estratégia de aquisição e medição dos dados

Esta atividade consiste na definição do procedimento que será utilizado para a medição de cada atributo de avaliação. Para isso, são identificadas fontes de dados disponíveis, como documentação do componente, página da Internet. Quando o componente já tiver sido submetido a uma avaliação anteriormente, os resultados da avaliação podem ser utilizados como fonte de informação. No caso de atributos que devem ser observados através de testes, devem ser definidos os casos de testes a serem executados, e quais técnicas serão utilizadas.

3.4 Aquisição e medição de dados

Esta etapa consiste na execução dos testes e na análise das fontes de dados selecionadas, planejados na etapa anterior.

3.5 Análise dos resultados

Os dados levantados são interpretados e, seguindo os critérios de avaliação definidos anteriormente, chega-se ao resultado procurado. Para cada componente avaliado deve ser gerado um relatório, de forma permitir consultas em futuras avaliações.

Tabela 1: Entradas e saídas para as atividades do processo de avaliação

| Atividade | Entradas | Saídas |
|--------------------------------------|--|--|
| Definição do modelo de qualidade | <ul style="list-style-type: none"> ▪ Requisitos funcionais e não funcionais da aplicação ▪ Diretrizes de qualidade da organização ▪ Modelo de qualidade base ▪ Relação dos componentes avaliados | <ul style="list-style-type: none"> ▪ Modelo de qualidade final: contém os atributos que serão analisados durante a avaliação, e as métricas que serão utilizadas. |
| Definição dos critérios de avaliação | <ul style="list-style-type: none"> ▪ Requisitos funcionais e não funcionais da aplicação ▪ Modelo de qualidade final ▪ Relação dos componentes avaliados | <ul style="list-style-type: none"> ▪ Critérios de avaliação: valores mínimos aceitáveis para os atributos analisados, pesos para cada atributo. |

| | | |
|--|---|--|
| Definição da estratégia de aquisição e medição dos dados | <ul style="list-style-type: none"> ▪ Relação dos componentes avaliados ▪ Artefatos dos componentes avaliados ▪ Modelo de qualidade final | <ul style="list-style-type: none"> ▪ Plano de execução da avaliação: funcionalidades que serão analisadas para cada componente, orientações para a execução das análises e testes, casos de testes. |
| Aquisição e medição de dados | <ul style="list-style-type: none"> ▪ Plano de execução da avaliação | <ul style="list-style-type: none"> ▪ Resultados medidos: valores obtidos a partir das análises e testes realizados, referências às origens dos dados, descrição dos testes executados. |
| Análise dos resultados | <ul style="list-style-type: none"> ▪ Resultados medidos ▪ Critérios de avaliação | <ul style="list-style-type: none"> ▪ Relatório final da avaliação: quais componentes foram aprovados para o uso. |

4. Execução do processo de avaliação

Esta seção apresenta os resultados de um experimento em que foi aplicado o processo de avaliação proposto.

O processo foi utilizado para a avaliação de quatro componentes na tecnologia Java para a geração de gráficos, tendo como objetivo a sua utilização em um projeto real, em que se está desenvolvendo um sistema com interface baseada em páginas de Internet. Os componentes A e B possuem licença comercial e os componentes C e D têm código aberto. Para os testes, foram utilizadas versões gratuitas de avaliação dos componentes A e B.

As subseções a seguir descrevem como foram realizadas cada uma das fases deste caso de avaliação e seus resultados.

4.1 Definição do modelo de qualidade

O modelo de qualidade final utilizado no experimento está detalhado na Tabela 2 **Erro! Fonte de referência não encontrada.** O modelo tem como base três características do modelo da ISO/IEC 9126: funcionalidade, usabilidade, eficiência e portabilidade. A confiabilidade foi deixada de fora por não ser muito relevante para a categoria de componente avaliada. A manutenibilidade não foi considerada, pois o objetivo é utilizar os componentes na forma disponibilizada pelos seus fornecedores, sem a realização de ajustes.

Tabela 2: Modelo de qualidade para componentes de geração de gráficos

| Fator de qualidade/ Atributo | Básico/ Derivado | Runtime/ ciclo de vida | Métrica | Escopo |
|---------------------------------|---------------------|---------------------------|---------------------------------------|--------|
| <i>Funcionalidade</i> | | | | |
| Configuração da aparência | Derivado | - | Codificado (Alta/ Média/ Baixa) | Função |
| Configuração de cores | Básico | Runtime | Presença | Função |
| Configuração de fonte | Básico | Runtime | Presença | Função |

| | | | | |
|---|----------|---------------|-----------------|-----------------------|
| Configuração de tamanho | Básico | Runtime | Presença | Função |
| <i>Usabilidade</i> | | | | |
| Tempo para configuração inicial do componente | Básico | Runtime | Valor (minutos) | Contexto (Componente) |
| Tempo para a geração de protótipo | Básico | Runtime | Valor (minutos) | Contexto (Função) |
| Exemplos demonstrativos na documentação | Básico | Ciclo de vida | Presença | Função |
| Proporção das funções com exemplos | Derivado | - | Taxa | Componente |
| <i>Eficiência</i> | | | | |
| Tempo de resposta | Básico | Runtime | Valor (ms) | Contexto (Função) |
| Tamanho do arquivo gerado | Básico | Runtime | Valor (kbytes) | Contexto (Função) |
| Tempo de processamento X quantidade de dados | Derivado | - | Curva | Contexto (Função) |
| <i>Portabilidade</i> | | | | |
| Compatibilidade com aplicações web | Básico | Runtime | Presença | Componente |
| Compatibilidade com aplicações desktop | Básico | Runtime | Presença | Componente |

A seleção dos atributos foi feita de forma a obter exemplos para os diferentes casos apresentados na seção 2 deste artigo. Existem também atributos específicos para a categoria de produto avaliada, referentes à capacidade de configuração da aparência dos gráficos gerados.

4.2 Definição dos critérios de avaliação

Nesta fase foi definido como os valores que serão obtidos na fase de aquisição e medição serão utilizados durante a análise de resultados. Os resultados são apresentados na Tabela 3 **Erro! Fonte de referência não encontrada..** A cada atributo foi associado:

- **Peso de importância:** Um valor numérico entre 1 a 5, escolhido arbitrariamente, representando a importância do atributo na avaliação;
- **Regra para a pontuação:** Uma regra estabelecendo como o valor obtido na medição será transformado em uma pontuação entre 0 a 1;
- **Condições mínimas:** Indica quais os valores aceitáveis mínimos procurados entre os componentes. Se um componente não obedecer a esse requisito, ele é automaticamente retirado da avaliação.

Tabela 3: Critérios de avaliação para componentes geradores de gráficos

| Fator de qualidade/ Atributo | Peso | Regra para a pontuação | Condições mínimas |
|---|-------------|---|------------------------------|
| Funcionalidade | | | |
| Configuração da aparência | 5 | Alta: 1; Média: 0,5; Baixa: 0 | - |
| Configuração de cores | 0 | Usar “Configuração da aparência” | - |
| Configuração de fonte | 0 | Usar “Configuração da aparência” | - |
| Configuração de tamanho | 0 | Usar “Configuração da aparência” | Sim |
| Usabilidade | | | |
| Tempo para configuração inicial do componente | 2 | Menor valor recebe 1, demais recebem pontos proporcionais ao valor. | - |
| Tempo para a geração de protótipo | 3 | Menor valor recebe 1, demais recebem pontos proporcionais ao valor. | - |
| Exemplos demonstrativos na documentação | 0 | Usar “Proporção das funções com exemplos”. | - |
| Proporção das funções com exemplos | 5 | Pontos = valor / 100 | - |
| Eficiência | | | |
| Tempo de resposta | 5 | Menor valor recebe 1, demais recebem pontos proporcionais ao valor. | - |
| Tamanho do arquivo gerado | 3 | Menor valor recebe 1, demais recebem pontos proporcionais ao valor. | - |
| Tempo de processamento X quantidade de dados | 2 | Pontuação proporcional ao coeficiente da reta. | - |
| Portabilidade | | | |
| Compatibilidade com aplicações web | 0 | Este valor é eliminatório, mas não influenciará a pontuação final. | Sim |
| Compatibilidade com aplicações desktop | 0 | Este valor é apenas informativo. | - |

4.3 Definição da estratégia de aquisição e medição de dados

Para os atributos cuja leitura depende de função, foram selecionadas as funções de geração de gráficos de barras, gráficos de linha e gráficos de pizza.

Todos os testes foram realizados em um laptop com processador de velocidade nominal de 1,6 GHz, com 2 Gb de memória RAM e Windows XP Professional instalado. O container web utilizado é o Tomcat versão 5.5.17.

Será também utilizada toda a documentação disponibilizada por cada fornecedor, em seus respectivos sites na Internet.

4.4 Aquisição e medição de dados

Os testes foram realizados utilizando o ambiente definido na fase anterior. Para cada componente foram construídos três protótipos, um para cada tipo de gráfico definido na fase anterior.

Os resultados de algumas das leituras estão representados na Tabela 4, Tabela 5 e na Figura 3. A Tabela 5 contém dados específicos para os protótipos de gráficos de linha. Já a Figura 3 compara a variação do tempo de resposta dos quatro componentes em relação ao número de dados utilizados para a geração de gráficos de linha.

Tabela 4: Resultados obtidos para atributos com escopo de componente

| Fator de qualidade/ Atributo | Comp A | Comp B | Comp C | Comp D |
|---|--------|--------|--------|--------|
| <i>Usabilidade</i> | | | | |
| Tempo para configuração inicial do componente | 46 min | 54 min | 79 min | 55 min |
| Proporção das funções com exemplos | 100% | 100% | 33% | 33% |

Tabela 5: Resultados obtidos para experimentos realizados com gráficos em linhas

| Fator de qualidade/ Atributo | Comp A | Comp B | Comp C | Comp D |
|---|--|---|---|---|
| <i>Funcionalidade</i> | | | | |
| Configuração da aparência | Média | Média | Baixa | Alta |
| Configuração de cores | Sim | Sim | Não | Sim |
| Configuração de fonte | Não | Não | Não | Sim |
| Configuração de tamanho | Sim | Sim | Sim | Sim |
| <i>Usabilidade</i> | | | | |
| Tempo para a geração de protótipo | 15 min | 15 min | 40 min | 33 min |
| Exemplos demonstrativos | Sim | Sim | Não | Não |
| <i>Eficiência</i> | | | | |
| Tempo de resposta (10/ 100/ 250/ 500/ 750/ 1000) (ms) | 29,6 ± 2,92/ 50,1 ± 4,96/ 58,0 ± 14,00/ 110,9 ± 17,48/ 137,5 ± 20,10/ 148,1 ± 31,72 | 61,0 ± 11,20/ 90,3 ± 19,24/ 123,5 ± 14,20/ 135,8 ± 18,24/ 156,2 ± 21,80/ 187,5 ± 19,00 | 125,2 ± 12,68/ 195,2 ± 12,44/ 303,3 ± 7,56/ 503,1 ± 54,76/ 767,3 ± 17,56/ 1120,3 ± 68,42 | 92,1 ± 2,62/ 118,7 ± 7,56/ 137,3 ± 4,92/ 149,9 ± 9,30/ 177,9 ± 9,70/ 229,9 ± 54,02 |
| Tamanho do arquivo gerado (10/ 100/ 250/ 500/ 750/ 1000) (KB) | 3,3/ 11,9/ 18,9/ 31,4/ 33,9/ 34,1 | 8,9/ 21,0/ 30,3/ 35,8/ 34,7/ 32,3 | 8,4/ 23,9/ 31,3/ 31,7/ 34,7/ 27,4 | 6,1/ 15,2/ 25,2/ 32,0/ 35,7/ 35,5 |

| | | | | |
|--|------------------------|------------------------|------------------------|------------------------|
| Tempo de processamento X quantidade de dados | $y = 0,1255x + 34,435$ | $y = 0,1133x + 76,432$ | $y = 0,9772x + 77,317$ | $y = 0,1219x + 97,956$ |
| <i>Portabilidade</i> | | | | |
| Compatibilidade com aplicações Web | Sim | Sim | Sim | Sim |
| Compatibilidade com aplicações Desktop | Sim | Sim | Sim | Não |

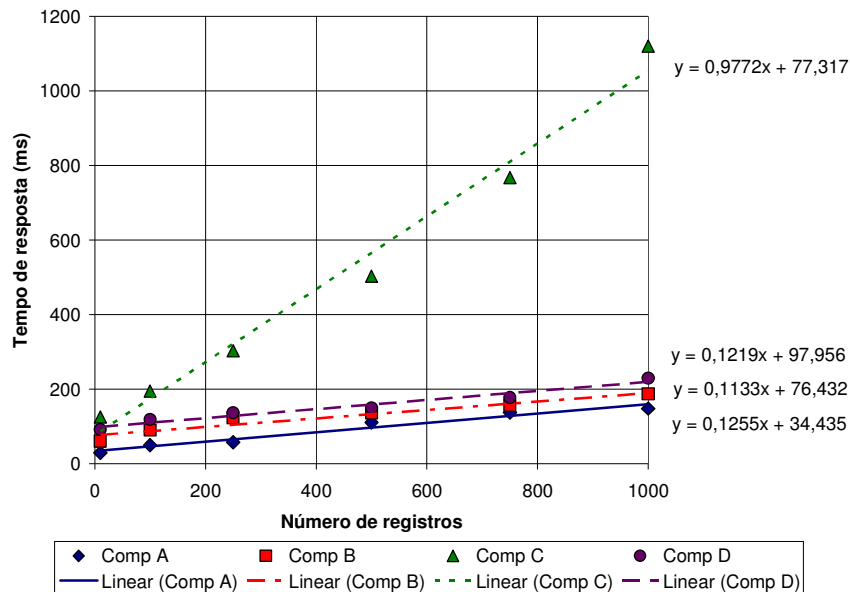


Figura 3: Tempo de resposta X número de registros para a geração de gráficos em linha

4.5 Análise dos resultados

Os dados obtidos durante a fase anterior foram utilizados para a geração dos pontos, de acordo com os critérios de avaliação definidos na segunda etapa do processo. A pontuação que cada componente recebeu em cada atributo está na Tabela 6 **Erro! Fonte de referência não encontrada.**

Utilizando a pontuação obtida na avaliação, conclui-se que o componente A foi o que obteve melhores resultados. O componente B apresentou resultados próximos para funcionalidade e usabilidade, porém o destaque do componente vencedor foi na eficiência.

Comparando separadamente os dois componentes de código livre, o componente D apresentou um desempenho bastante melhor que o componente C, aproximando-se bastante dos resultados obtidos pelos componentes comerciais. Além disso, sua capacidade de configuração de aparência foi a melhor entre todos os componentes avaliados.

Tabela 6: Pontuação final dos componentes avaliados no experimento

| Fator de qualidade/ Atributo | Comp A | Comp B | Comp C | Comp D |
|---|------------------|------------------|-----------------|----------------|
| <i>Funcionalidade</i> | | | | |
| Configuração da aparência | 0,5 (2,5) | 0,5 (2,5) | 0 (0) | 1 (5) |
| <i>Usabilidade</i> | | | | |
| Tempo para configuração inicial do componente | 1 (2) | 0,9 (1,8) | 0,6 (1,2) | 0,9 (1,8) |
| Tempo para a geração de protótipo | 1 (3) | 1 (3) | 0,4 (1,2) | 0,5 (1,5) |
| Proporção das funções com exemplos | 1 (5) | 1 (5) | 0,3 (1,5) | 0,3 (1,5) |
| <i>Eficiência</i> | | | | |
| Tempo de resposta | 1 (5) | 0,7 (3,5) | 0,2 (1) | 0,6 (3) |
| Tamanho do arquivo gerado | 1 (3) | 0,7 (2,1) | 0,7 (2,1) | 0,8 (2,4) |
| Tempo de processamento X quantidade de dados | 0,9 (1,8) | 1 (2) | 0,1 (0,2) | 0,9 (1,8) |
| PONTUAÇÃO FINAL | 22,3 / 25 | 19,9 / 25 | 7,2 / 25 | 17 / 25 |

5. Conclusões e trabalhos futuros

O processo de avaliação de componentes proposto pode ser utilizado como base para a avaliação em diferentes contextos. O modelo de qualidade é o elemento central da avaliação. Mesmo existindo diferentes estudos para a elaboração de modelos de qualidade especiais para componentes de software, o processo proposto foi elaborado com o objetivo de ser adequado a esses estudos.

O modelo de qualidade utilizado na ilustração do uso do processo obteve resultados úteis, que permitiram identificar o componente com melhor desempenho entre as opções selecionadas. Em oportunidades futuras, os mesmos dados podem ser aproveitados para comparar com outros componentes, desde que o procedimento para a avaliação dos novos candidatos seja realizado sob condições semelhantes às utilizadas neste experimento. O próprio modelo pode passar por modificações, incluindo atributos referentes às outras características da norma ISO/IEC 9126.

Como trabalho futuro, será estudado como incorporar no processo metodologias para a escolha e avaliação de atributos de avaliação e métricas [Port 2004], [Bertoa 2005], [Torchiano 2003], [Franch 2003]. Além disso, o processo de definição de modelos de qualidade será mais detalhado, com o objetivo de identificar diferentes níveis de especialização.

Referências

Alvaro, A., Meira, S. R. L., Almeida, E. S. (2005) Quality Attributes for a Component Quality Model. In the *10th International Workshop on Component-Oriented*

- Programming (WCOP) in Conjunction with the 19th European Conference on Object Oriented Programming (ECOOP)*, Glasgow, 2005.
- Alvaro, A., Meira, S. R. L. (2005b) Component Certification: A Component Quality Model. In: *III Workshop de Teses e Dissertações em Qualidade de Software*, Porto Alegre, 2005.
- Bertoa, M. F. and Vallecillo, A. (2002) Quality Attributes for COTS Components. In *Proc. of the 6th ECOOP Workshop on Quantitative Approaches in Object Oriented Software Engineering (QAOOSE 2002)*, Malaga, Spain, June 2002.
- Bertoa, M. F. and Vallecillo, A. (2005) Usability Indicators for Software Components. In *Proc. of the 9th ECOOP Workshop on Quantitative Approaches in Object Oriented Engineering (QAOOSE 2005)*, Glasgow, United Kingdom, July 25, 2005.
- Carvalho, J. P., Franch, X. (2006) Extending the ISO/IEC 9126-1 Quality Model with Non-Technical Factors for COTS Components Selection. In *Proceedings of the 2006 International Workshop on Software Quality*. Shanghai, China, 2006. Pages 9-14. ACM Press.
- Franch, X. and Carvalho, J. P. (2003). Using Quality Models In Software Package Selection. In *Software*, vol. 20, no. 1, Jan-Feb 2003, pages 34-41. IEEE Computer Society.
- Goulão, M., Abreu, F. B. (2004). Software Components Evaluation: an Overview. In *Atas da 5ª Conferência da APSI*, Lisboa, Novembro 2004
- Goulão, M., Abreu, F. B. (2004b). Formalizing metrics for COTS. In *Proc. of the International Workshop on Models and Processes for the Evaluation of COTS Components (MPEC'04) at the ICSE'2004*, Edimburgh, Scotland. May, 2004.
- Meyer, B. (2003) The Grand Challenge of Trusted Components. In the *25th IEEE International Conference on Software Engineering (ICSE 2003)*, USA, 2003.
- Port, D. and Chen, Z. H. (2004) Accessing COTS Assessment: How Much Is Enough? In *Proc. of the ICCBSS 2004*, February, 2004, pages 183-198.
- Torchiano, M., Jaccheri, L., Sørensen, C. F. and Wang, A. I. (2002). COTS Products Characterization. In *Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (SEKE'02)*, Ischia, Italy, 2002, pages 335-338. ACM Press.
- Torchiano, M. and Jaccheri, L. (2003) Assessment of Reusable COTS Attributes. In *Proc. of the 2nd Int'l Conference on COTS Based Software Systems (ICCBSS'2003)*, Ottawa, Canada, February 10-12, 2003, pages 219-228.
- Voas, J. (1999) Certifying software for high-assurance environments. In *Software*, vol. 16, no. 4, July-Aug. 1999, pages 48-54. IEEE Computer Society.
- Washizaki, H., Yamamoto, H., Fukazawa, Y. (2003) A Metrics Suite for Measuring Reusability of Software Components. In: *Proceedings of the Ninth International Software Metrics Symposium (METRICS'03)* 2003.