

Avaliação da Manutenibilidade da Arquitetura de Software

Alexandre M. L. Vasconcelos, Diego de Azevedo Ribeiro

Centro de Informática – Universidade Federal de Pernambuco (UFPE)

Caixa Portal 7851 – CEP 50732 – 970 – Recife – PE - Brasil

{amlv,dar}@cin.ufpe.br

Abstract. The objective of this paper is to define maintainability process evaluation of a software architecture being based on ISO 9126 quality characteristics and sub-characteristics. Also they had been defined metric that must be collected during the design phase of the software life cycle.

Resumo. O objetivo deste trabalho é definir um processo de avaliação do fator de manutenibilidade de uma arquitetura de software baseando-se nas características e sub-características de qualidade definidas na norma 9126 da ISO. Também foram definidas métricas que devem ser coletadas durante a fase de projeto do ciclo de vida do software.

1 Introdução

Devido ao grande crescimento de indústrias e empresas desenvolvedoras de software, a concorrência entre os produtos finais gerados por estas empresas está cada vez mais acirrada. Para garantir o uso de seu produto as empresas estão valorizando iniciativas que permitam que o seu resultado final possua elementos diferenciadores dos outros.

A qualidade do software, assunto estudado pela área de Engenharia de software, está cada vez mais sendo procurada por essas empresas com o intuito de valorizar os seus produtos e serviços no mercado de software [GRUNSKÉ et al 2002]. A qualidade é uma característica importante, pois ela ajuda a garantir a satisfação do cliente fazendo com que este procure a empresa mais vezes. Além disso, a produção de produtos com qualidade reduz o custo da empresa no momento da manutenção do software.

Existem grandes estudos sendo executados nesta área, porém muitos pontos ainda não foram esclarecidos ou avaliados. Uma das grandes dificuldades para a avaliação da qualidade do produto de software é que grande parte dos testes só pode ser executada após a finalização do software. Isto significa um alto custo e um grande esforço de mudança caso a qualidade desejada não seja atingida.

Para evitar estas situações, cada vez mais estão surgindo metodologias e técnicas que auxiliam tanto no processo de desenvolvimento do software como também para a avaliação dos artefatos gerados. Baseando-se nesta idéia, foram criados vários trabalhos para avaliar a qualidade do produto de software através do estudo de sua arquitetura. Através da arquitetura do sistema, é possível prever alguns dos possíveis comportamentos do software antes de sua conclusão, além de poderem ser observadas características importantes como coesão, que influencia na manutenção do produto.

A arquitetura é basicamente definida por requisitos não funcionais [BACHMANN et al 2005]. Por isso, ela é capaz de responder sobre o comportamento final do software relacionado às características como escalabilidade, segurança, manutenibilidade e desempenho [BENGTSSON 1998] [BENGTSSON et al 2001] [MENDES 2002], sendo inclusive considerada a chave para atingir as exigências desses requisitos [BARBACCI et al 2003] [BARBACCI et al 2005]. Utilizando-se deste fato,

uma avaliação qualitativa da arquitetura pode prever problemas no software antes dele ser finalizado. Isto diminuiria o custo para adequar o sistema aos padrões de qualidade.

Existem várias técnicas de avaliação e análise arquiteturas. Algumas técnicas avaliam alguns detalhes da arquitetura fazendo comparativos entre duas propostas arquiteturas para definir qual se aplica melhor aos requisitos. Outras avaliam, de uma forma global, uma única arquitetura para determinar se a arquitetura proposta possui os atributos de qualidade necessários. Existem até mesmo, estudos comparativos entre estas técnicas. A unificação de atividades dessas técnicas pode resultar em um processo de avaliação arquitetural completo.

Um fator importante a ser citado é que o custo para reparar falhas nos sistemas vai aumentando à medida que o ciclo de desenvolvimento do software vai atingindo seu final. Por este motivo, a padronização de um processo para a avaliação arquitetural, pode reduzir o custo na produção, já que através deste processo, é possível detectar e repara problemas em fases iniciais do ciclo de vida de um software. Além disso, o grau de qualidade poderia ser garantido para todos os produtos, nivelando qualitativamente os softwares produzidos.

O documento está organizado da seguinte forma: a Seção 2 descreve alguns processos para avaliação de arquitetura. A Seção 3 define o processo para avaliação da qualidade de manutenibilidade da arquitetura. Na Seção 4, é ilustrada a utilização do processo para a execução de uma avaliação de arquitetura. Por fim, é feita uma conclusão avaliando todos os marcos do trabalho ressaltando os pontos positivos e negativos deste documento.

2 Avaliação da Arquitetura de Software

No decorrer dos últimos anos, as funcionalidades críticas dos sistemas vão lentamente se transferindo do hardware para o software. Juntamente com esta transformação a complexidade dos sistemas vai aumentando [GRUNSKE et al 2002]. O desafio não é mais prover os atributos funcionais, mas sim garantir os atributos de qualidade, pois estes requerem uma atenção explícita, para que, no final, seja atingido um nível satisfatório.

A arquitetura deve especificar um mapeamento com os requisitos não funcionais e atributos de qualidade, pois o sistema será fundamentalmente afetado pelas decisões tomadas no desenvolvimento da arquitetura do software. Os requisitos de qualidade direcionam a arquitetura do software da mesma forma que as atividades centradas na arquitetura definem o ciclo de vida do software.

Apesar de serem propostos alguns métodos de avaliação da qualidade da arquitetura [BARBACCI et al 2003] [BENGTSSON 1998], eles possuem quase sempre os mesmos objetivos: analisar o impacto futuro de decisões atuais nos requisitos de qualidade, encontrar pontos sensíveis na arquitetura cuja mudança poderia causar problemas de qualidade do software e, por último, mas não menos importante, elaborar *tradeoffs* nas decisões de projeto de arquitetura que influenciem mais de um atributo de qualidade.

Outro fator importante na avaliação da arquitetura é que os atributos de qualidade, por si só, não contêm informações o bastante para influenciar o sistema. Por exemplo, “o sistema deve ser modificável”, não agrega informações ao sistema, pois todo o sistema é modificável com respeito a um conjunto de mudanças e não com respeito a outros conjuntos.

Uma solução para este problema de descrição dos atributos de qualidade é a utilização de cenários, como meio de caracterizar melhor esses requisitos. Os cenários também são uma poderosa maneira para representar as visões e conceitos que os *stakeholders* possuem desses atributos. Vários processos de avaliação utilizam a idéia de cenários, entre eles podem ser citados SAAM e QAW [MENDES 2002] [BARBACCI et al 2005]. O processo proposto também é baseado em cenários, permitindo simular as necessidades e as mudanças que podem ocorrer no software.

3 Processo de Avaliação da Qualidade da Arquitetura

A avaliação de arquitetura tem como objetivo avaliar se a arquitetura projetada para o sistema é capaz de atender às funcionalidades desejadas e respeita requisitos não funcionais de manutenibilidade focando nas sub-características de modificabilidade, conformidade e analisabilidade, de acordo com a definição das características de qualidade da ISO 9126 [ISO/9126 2002]. As verificações da documentação da arquitetura, da consistência entre o diagrama arquitetural e do Programa-Fonte do sistema implementado também são realizadas neste módulo.

A modificabilidade está relacionada ao custo para fazer alterações. A analisabilidade por outro lado, destina-se a identificar qual o custo para entender os componentes da arquitetura. A conformidade está sendo avaliada no instante em que existem atividades para verificar a coerência com os padrões arquiteturais adotados.

3.1 Artefatos do Processo

No decorrer do processo serão gerados e utilizados vários artefatos. Entre esses artefatos existem aqueles que são comuns a todos os processos de desenvolvimento e servem de entrada para as atividades de avaliação. Outros artefatos foram criados durante a execução das atividades desse processo. Os artefatos são os seguintes:

Especificação de Requisitos de Software - É o documento de requisitos do software

Programa-Fonte - O programa-fonte é o código fonte do sistema, ou seja, é o código dos componentes implementados no sistema.

Especificação de sistema e projeto de sistema (Documento de Arquitetura) - Este documento define quais são os padrões utilizados na arquitetura, quais são os principais componentes e como eles estão interligados.

Cenários - Documentos contendo descrições das atividades do sistema cuja necessidade de avaliação foi verificada.

Documentos de Falhas e Inconsistências - Documento contendo todas as inconsistências entre a implementação e a documentação da arquitetura e falhas com relação ao padrão de projeto adotado.

Documento de Avaliação dos Cenários - Documento contendo o resultado final da avaliação dos cenários.

Relatório Final do Projeto - Relatório contendo os cenários utilizados, ferramentas adotadas, identificação dos documentos utilizados para realização dos testes e um sumário das principais descobertas.

3.2 Atividades do Processo

As atividades do processo foram definidas através de pesquisas e outros métodos de avaliação utilizados, além de considerar as características do processo de avaliação da qualidade definido na ISO 14598 [ISO/14598 2002]. Como pode ser observada na Figura 1 a base deste processo é a avaliação da qualidade do software através da elaboração de cenários. Os pentágonos azuis representam as atividades que devem ser executadas pelo avaliador para determinar o nível de qualidade da arquitetura. As folhas representam os artefatos que servem como entrada e saída das atividades.

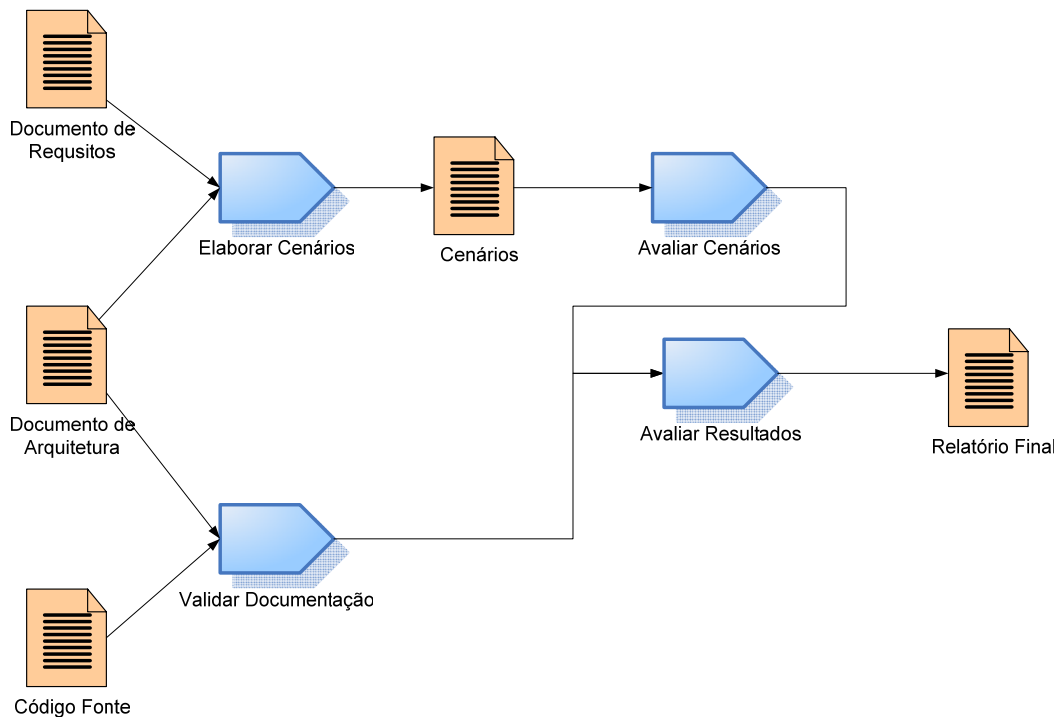


Figura 1 – Processo para avaliação da qualidade arquitetura

Foram definidas 4 atividades: validar documentação, elaborar cenários, avaliar cenários e avaliar resultados.

A validação da documentação tem a finalidade de verificar a consistência entre os documentos da arquitetura e a arquitetura projetada, como também a implementação real da arquitetura.

A atividade “Elaborar Cenário” visa à criação de cenários de testes para avaliar a arquitetura. É necessária a elaboração de dois tipos de cenários: funcionais e de modificabilidade. Para cada funcionalidade elicitada através da consulta ao documento de Especificação de Requisitos de Software, o avaliador deve construir um cenário funcional e determinar quais os componentes da arquitetura estão envolvidos nesta funcionalidade. Para cada possível modificação de requisitos elicitada através da entrevista ao gerente e de consultas ao documento de Especificação de Requisitos de Software, o avaliador deve construir um cenário de modificabilidade determinando quais componentes sofrerão impacto com a construção desta mudança.

Após a elaboração dos cenários é iniciada a atividade de avaliação de cenários. Nesta atividade é feita a análise dos cenários criados e são utilizadas as métricas de cenário para determinar o resultado dos cenários construídos.

Finalmente, durante a atividade de avaliação dos resultados são consolidados os dados das avaliações executadas e é elaborado um relatório final, ressaltando os pontos fortes e fracos da arquitetura com relação ao requisito não funcional de manutenibilidade.

3.3 Métricas utilizadas

Durante a elaboração do processo foram definidas duas métricas, cujas escalas foram definidas entre 0 e 100, que devem ser utilizadas durante a sua execução. Uma métrica determina os desvios entre os documentos e a implementação. A outra métrica determina o grau de modificabilidade de cada componente da arquitetura.

A primeira métrica, resultado dos desvios, é obtida a partir de dois valores: o valor gerado pelas inconsistências de documentação e padrões, e o valor gerado pelas inconsistências de implementação.

O primeiro valor é o número de disparidades entre as características listadas no documento de arquitetura e o diagrama de arquitetura do projeto, dividido pelo número de características citadas no documento. O segundo valor é o número de disparidades entre o projeto da arquitetura e a implementação do software dividido pelo número de componentes da arquitetura. A nota final é obtida pela seguinte fórmula $100 - (\text{valor1} + \text{valor2}) * 100$.

A notificação de modificabilidade é baseada na informação de dois pesos para cada componente da arquitetura. O primeiro peso (peso funcional) está relacionado à quantidade de funcionalidades que o componente possui. Isso deve ser identificado levando em consideração os cenários de funcionalidades.

Todos os componentes devem iniciar com o peso funcional igual a 0. Para cada envolvimento deste componente no fluxo de um cenário funcional deve ser adicionado mais 1 no peso funcional do componente. O segundo peso (peso de modificação) está relacionado às chances do componente sofrer modificação. O peso de modificação de cada componente também é iniciado com 0. Para cada envolvimento do componente nos cenários de modificabilidade deve ser somado mais 1 neste peso. Após o cálculo dos pesos de todos os componentes presentes nos cenários é necessário calcular o *score* de cada componente. Este valor é o resultado da multiplicação entre o peso funcional e o peso de modificação. Com os *scores* calculados utiliza-se a equação na Tabela 1 para determinar a dificuldade de modificabilidade.

Tabela 1 – Equação para grau de modificabilidade da arquitetura

$1 - ((\text{Soma dos } \textit{scores} \text{ de todos os componentes}) / (\text{Quantidade de cenários funcionais testados} \times \text{Quantidade de cenários de modificabilidade testados} \times \text{numero de componentes dos cenários})) \times 100.$

4 Utilizando o Processo de Avaliação

Nesta seção será apresentada uma avaliação utilizando o processo proposto. Esta simulação será executada sobre um portal WEB. Este portal é um novo sistema que está sendo produzido com o intuito de centralizar em um único local, informações de todos os fornecedores de um supermercado. Através deste sistema será possível cadastrar e consultar dados de todos os fornecedores além de possibilitar a solicitação de produtos fornecidos, auxiliando no gerenciamento e facilitando o acompanhamento do histórico do negócio dos usuários.

O sistema será acessado por centenas de usuários ao mesmo tempo. Os relatórios gerados serão exibidos em páginas html e o usuário poderá definir os parâmetros das consultas. Os usuários também poderão observar os históricos de suas transações de cadastramento feitas no portal, possibilitando a execução de auditorias caso seja necessário.

4.1 Cenários da Arquitetura

A arquitetura desenvolvida para este sistema pode ser observada na Figura 2. O ActionController é o componente responsável por controlar todos os Actions. O Action é responsável por tratar os dados provenientes da tela e repassá-los para a camada de negócio. A Fachada serve como ponto único para a camada de negócio do software. O Controller executa todas as regras de negócio antes dos dados serem armazenados. Os componentes IRepository e Repository tem a finalidade de persistir os dados. O Repository utiliza o GeneralRepository que executa a persistência em banco de dados. O último componente é o engineReport responsável pela geração dos relatórios do sistema.

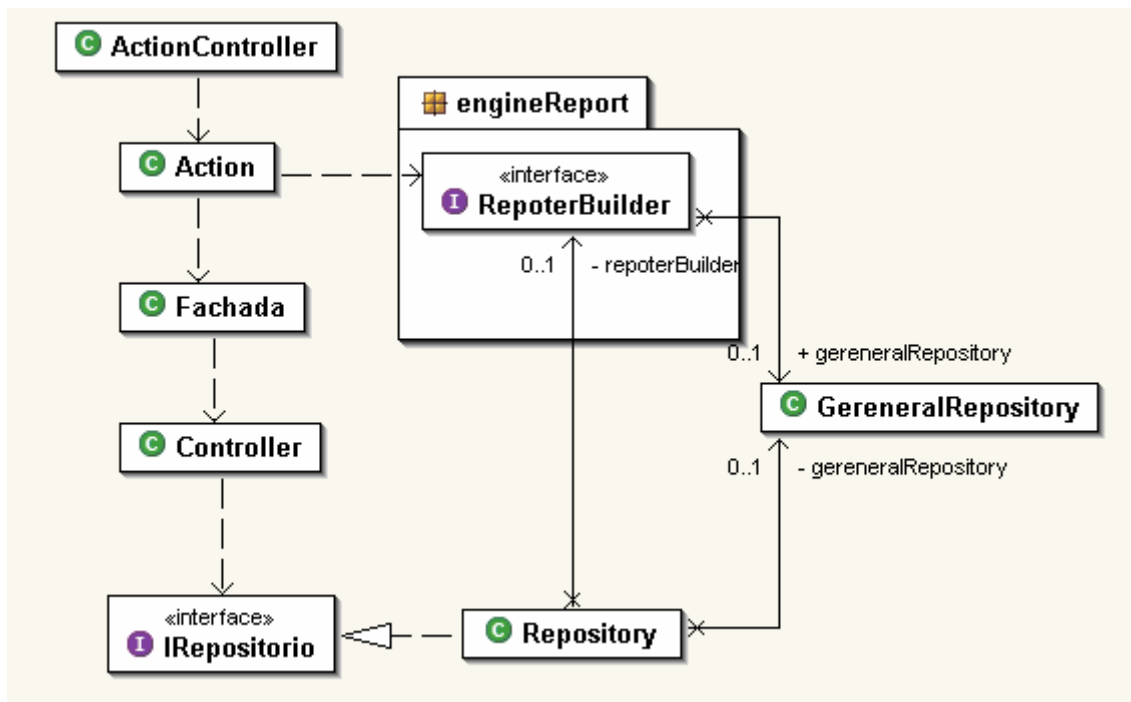


Figura 2 – Arquitetura do sistema

4.2 Avaliação de Modificabilidade

Como esta avaliação tem a finalidade de exemplificar o processo, e não de definir a qualidade do produto, foram criados cenários para apenas algumas funcionalidades do software. Além disso, só serão exibidos os resultados das atividades de elaborar e avaliar cenários, pois estas são as atividades centrais do processo. Foram definidos dois cenários funcionais e dois cenários de modificabilidade. As especificações destes cenários podem ser vistas abaixo:

Cenário Funcional 1

Descrição: Cadastrar novo produto na base de dados.

Componentes envolvidos: ActionController, Action, Fachada, Controller, IRepositorio, Repository e GeneralRepository

Cenário Funcional 2

Descrição: Consultar relatório de novos produtos

Componentes envolvidos: ActionController, Action, EngineReport e GeneralRepository

Cenário Modificabilidade 1

Descrição: Os relatórios devem ser arquivos pdf e não mais html.

Componentes envolvidos: ActionController, Action, EngineReport

Cenário Modificabilidade 2

Descrição: Dados deverão ser armazenados em arquivo

Componentes envolvidos: Repository, EngineReport

Baseando-se nestes cenários os pesos foram coletados e os *scores* dos componentes foram calculados. Finalmente os valores foram consolidados gerando a nota de modificabilidade do software. Estes dados podem ser observados na Tabela 2. Como pode ser visto o grau de modificabilidade deste software, levando em consideração estes cenários é de 78. É importante ressaltar que este resultado deve estar associado a um contexto de qualidade para definir se o valor obtido é satisfatório ou é necessária a execução de um plano de melhoria da qualidade.

Tabela 2 – Dados da avaliação

ActionController	2
Action	2
Fachada	0
Controller	0
IRepositorio	0
Repository	1
GeneralRepository	0
EngineReport	2
$(1 - (7 / (2 \times 2 \times 8))) \times 100$	78

5 Conclusão

A qualidade não está apenas nas funcionalidades. Ela também deve ser levada em consideração no esforço gasto para manter o software sempre atualizado com as mudanças requisitadas. Devido a isso existe a necessidade de averiguar a facilidade de manutenção do software durante todo o seu ciclo de vida.

A escolha da arquitetura como base do processo foi devido ao fato de estudos e pesquisas provarem que grande parte dos problemas de qualidade de um software são conseqüências de problemas originados na fase de projeto do software.

Apesar de existirem vários estudos sobre o relacionamento entre atributos de qualidade e a arquitetura de software, ainda existem espaços a serem preenchidos nessa

relação. Atualmente, não existem mapeamentos perfeitos entre as características presentes na arquitetura para atributos qualitativos do produto de software final.

A sugestão de uma maneira para estimar o grau de dificuldade de manutenção, também é de grande relevância, uma vez que a maior parte das métricas consideradas para avaliar essa característica é baseada em dados provenientes da execução da própria atividade de manutenção. Assim, com um dado preventivo é possível tomar algumas precauções para auxiliar na redução de custos de um software durante sua fase de manutenção.

Como um passo complementar ao processo definido neste documento, é necessário a criação de novas métricas de avaliação da arquitetura, baseando-se em outras características da ISO 9126, com o intuito de abranger um escopo maior de qualidade. Além disso, é necessário projetar a inclusão deste processo no ciclo de vida de desenvolvimento do software para que seja possível executar um gerenciamento mais eficiente da arquitetura durante sua construção.

Referências

- [BACHMANN et al 2005] BACHMANN, F.; BASS, L.; KLEIN, M. **Illuminating the Fundamental Contributors to Software Architecture Quality**, Technical report CMU/SEI-2002-TR-025, Software Engineering Institute, Carnegie Mellon University, 2002. Disponível em: < <http://www.sei.cmu.edu/publications/documents/02.reports/02tr025.html>>. Acesso em: 18 out. 2005.
- [BARBACCI et al 2005] BARBACCI, M.; BERGEY, J.; WOOD, W. **Using Quality Attribute Workshops to Evaluate Architectural Design Approaches in a Major System Acquisition: A Case Study (CMU/SEI-2000-TN-010, ADA395200)**. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. Disponível em: <<http://www.sei.cmu.edu/publications/documents/00.reports/00tn010.html>>. Acesso em 18 out. 2005
- [BARBACCI et al 2003] BARBARCCI, M. R. et al. **Quality Attributes Workshop, CMU/SEI-2003**. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003.
- [BENGTSSON et al 2001] BENGTSSON, P.; BOSCH, J., **Assessing Optimal Software Architecture Maintainability**, Proceedings of the Fifth European Conference on Software Maintenance and Reengineering (CSMR 2001), 2001.
- [BENGTSSON 1998] BENGTSSON, P., **Towards Maintainability Metrics on Software Architecture: An Adaptation of Object-Oriented Metrics**, First Nordic Workshop on Software Architecture (NOSA'98), Ronneby, 1998
- [GUPTA 1997] GUPTA, B. S. **A Critique of Cohesion Measures in the Object-Oriented Paradigm**, 1997. Disponível em: < <http://citeseer.nj.nec.com/cache/p:zSzzSzcs.mtu.edu/zSzpubzSzottzSzreportszSzmehra-thesis.pdf/gupta97critique.pdf> > Acesso em 18 out. 2005
- [GRUNSKÉ et al 2002] GRUNSKÉ, L.; NEUMANN, R., **Quality Improvement by integrating Non-Functional Properties in Software Architecture Specification**, Second Workshop on Evaluating and Architecting System dependability (EASY). California, U.S.A, 2002

- [MENDES 2002] MENDES, A. **Arquitetura de Software, Desenvolvimento orientado para arquitetura**, 1. ed.: Editora Campus, 2002.
- [SOMERVILLE 2003] SOMERVILLE, I., **Engenharia de Software**. Tradução de Maurício de Andrade. Revisão técnica Prof. Dr. Kechi Hiramã. 6. ed. São Paulo: Editora Afiliada, 2003
- [ISO/9126 2002] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, **9126 – Engenharia de Software – Qualidade do Produto**, 2002
- [ISO/14598 2002] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, **14598 – Engenharia de Software – Avaliação de Produto**, 2002