



**Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Curso de Engenharia de Software**

**Teste de software Sob a Perspectiva de Manutenibilidade com foco em
Analisabilidade**

**Gama - DF
Outubro de 2019**



Autores:

Ésio G. Pereira Freitas	17/0033066	100%
Fabiana L. V. Pfeilsticker Ribas	16/0005736	100%
Fernando Ribeiro Aguiar	14/0139281	100%
Lucas Dutra F. do Nascimento	17/0050939	100%
Mikhaelle de Carvalho Bueno	15/0018673	100%
Rafael M. Gomes Ferreira	16/0142369	100%
Rogério S. dos Santos Júnior	17/0021751	100%
Youssef M. Yacoub Falaneh	17/0024334	100%

Orientador:

Dr. Ricardo Ajax Dias Kosloski

Teste de software Sob a Perspectiva de Manutenibilidade com foco em Analisabilidade

Gama - DF

Outubro de 2019



Sumário

Sumário	2
Resumo	3
Lista de Siglas	3
Introdução	4
Referencial Teórico	4
Características	4
Definição de Manutenibilidade	5
Subcaracterísticas	5
Definição de Analisabilidade	5
Problemas Encontrados	6
Objetivos Gerais	6
Objetivos Específicos	7
Medidas e processos usados	7
Definição de Métricas	7
Métricas Internas de Analisabilidade	8
Gravação de atividade:	8
Prontidão da função de diagnóstico:	9
Métricas Externas de Analisabilidade	9
Capacidade de trilha de auditoria	9
Suporte à função de diagnóstico	9
Capacidade de análise de falhas	10
Eficiência na análise de falhas	10
Capacidade de monitoramento de status	10
Testes usados para verificação e validação	10
Conclusão	12
Referências	13



Resumo

A *manutenibilidade* de um software é uma das características mais importantes na qualidade do software. Neste trabalho é abordado o conceito de *manutenibilidade*, a capacidade do software de ser modificado, e suas subcaracterísticas, analisabilidade, estabilidade, alterabilidade e testabilidade, dando foco para o ponto de vista da *analisabilidade* do produto de software. A Analisabilidade é definida pela ISO 9126: 2001, 6.5.1 e tem o objetivo de descobrir as deficiências em estágio inicial ou identificar precocemente o local onde a falha ocorreu. A ISO 9126 define um conjunto de parâmetros com o objetivo de padronizar a avaliação da qualidade de software. O trabalho também traz medidas e processos que podem ser usados para medir a qualidade com foco em analisabilidade da manutenibilidade e quais tipos de testes podem ser realizados para verificação.

Palavras-chave: Teste de Software. Qualidade de Software. ISO/IEC 9126-1. Processos. Métricas de qualidade. Manutenibilidade. Analisabilidade. Verificação.

Lista de Siglas

Sigla	Significado
ISO/IEC 9126	Norma ISO para qualidade de produto de software
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
IEEE	The Institute of Electrical and Electronics Engineers



Introdução

A Manutenibilidade é uma das características mais significativas da qualidade e do ciclo de vida de um software. Muitas empresas investem cerca de 60 a 70% na correção e manutenção de softwares já existentes[1]. A maioria das empresas investem cerca de 70% em testes de manutenção de um software para controlar e acompanhar e melhorar a qualidade do mesmo[2]. Este trabalho visa verificar sob a perspectiva de teste de software e qualidade de software a característica da Manutenibilidade de um software sobre o ponto de vista da subcaracterística de Analisabilidade do produto de software. Além de apresentar possíveis formas de analisar um software sobre tais perspectivas. Para concretizar melhor a ideia do trabalho, serão fornecidos o conceito de manutenibilidade e suas subcaracterísticas. Posteriormente, será demonstrado a aplicação de cada conceito abordado dentro de um projeto de software, a fim de esclarecer a importância da mesma.

Referencial Teórico

Características

Como referencial teórico, neste trabalho foi utilizado como principal fonte de apoio a norma *ISO / IEC 9126-1* que descreve um modelo de qualidade para o produto de software em 6 características principais[3]:

- Funcionalidade;
- Confiabilidade;
- Usabilidade;
- Eficiência;
- Manutenibilidade;
- Portabilidade;



Definição de Manutenibilidade

Como especificado anteriormente, o foco deste trabalho será em *Manutenibilidade*. A definição de *Manutenibilidade* de acordo com o glossário de Engenharia de Software do IEEE é :

[...“*the ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a change environment.*”...],

ou seja, capacidade do 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 e não funcionais.

Subcaracterísticas

A característica de *Manutenibilidade* pode ser dividida em mais 4 sub características mensuráveis, são elas[4]:

- *Analisabilidade;*
- *Alterabilidade;*
- *Estabilidade;*
- *Testabilidade;*

Neste trabalho, especificamente será trabalhado a sub característica de *Analisabilidade*.

[“...*Métricas externas de manutenibilidade devem ser capazes de medir atributos tais como o comportamento do mantenedor, usuário, ou do sistema que contém o software, quando o software é mantido ou modificado durante os testes ou a manutenção...*”] (Comitê Brasileiro de Computadores e Processamento de Dados, 2003, p.10)

Definição de Analisabilidade

Analisabilidade é definida pela ISO 9126: 2001, 6.5.1, como: “Capacidade do produto de software de permitir diagnóstico de deficiência ou causas de falhas no software ou a identificação de partes a serem modificadas[5]. Para descobrir as deficiências em estágio



inicial ou identificação precoce do local onde a falha ocorreu é um esforço valioso para mitigar possíveis problemas.

Problemas Encontrados

Não é possível de ser medida diretamente a manutenibilidade de um software, mas com a definição de algumas características internas é possível analisar, controlar e melhorar a qualidade. Mas o que medir? Como testar? O que devemos testar no software para avaliar o nível de complexidade de manutenibilidade de um software? Algumas influências já podem ser observadas de antemão que possam impactar na analisabilidade do software como documentação, atualizada se possível, design, legibilidade do código, complexidade do código entre outras boas práticas, disponibilidade dos desenvolvedores para esclarecimento, existência de controle de configuração de software entre outro.

Objetivos Gerais

De acordo com a ISO/IEC 9126, uma norma que, define um conjunto de parâmetros com o objetivo de padronizar a avaliação da qualidade de software, a qualidade de produto de software, é altamente influenciada pela qualidade dos processos de software. E para avaliar esses processos usamos métodos e métricas para medir, testar, entender, prever e aperfeiçoar os processos.

Dada a característica e a subcaracterística de qualidade descrita no tema definido, o trabalho consiste em abordar uma das características de qualidade em específico e apresentar formas de solucionar a seguinte questão: “Como testar um produto de software sob o foco da característica de Manutenibilidade?”.



Objetivos Específicos

Dentro do escopo deste trabalho, será abordado formas de como é possível testar a Manutenibilidade de um software com foco em analisabilidade, sob o ponto de vista do de produto de software.

A. Garvin em seu artigo “What Does “Product Quality” Really Mean”s descreve *Visão de Produto*[7] de forma a observar a relação entre a qualidade interna e a externa de um produto, pregando que boa qualidade interna influencia positivamente a qualidade externa;

A visão do produto serve para caracterizar o que se deseja medir, isso é especialmente útil para o controle de qualidade pois, ao dar noção do que se busca medir, auxilia a definir as perguntas e as métricas que serão levantadas para determinado objetivo dentro de um projeto ou organização.

Medidas e processos usados

Para medir a qualidade de um produto de software com foco em Manutenibilidade é necessário definir um processo para identificar os objetivos para os quais buscamos respostas e a partir deles definir questões que permitam a definição de métricas para se realizar a medição. Os resultados da medição servem para guiar a tomada de decisão e futuros incrementos.

Don Coleman e Dan Ash[10] analisaram cinco métodos para medição da manutenibilidade do software. Modelos hierárquicos, modelos de regressão polinomial, medida de complexidade agregada, análise de componentes principais e análise fatorial e os testes dos modelos indicavam que todos calculam pontuações de manutenibilidade razoavelmente precisas a partir de cálculos baseados em métricas simples.

Definição de Métricas

As métricas podem ser definidas como parâmetros de medição voltados a propriedades específicas de um produto de software, necessário para se avaliar e garantir a qualidade do mesmo. Elas podem ser divididas em dois tipos:



- *Interna:* As métricas internas podem ser aplicadas a um produto de software não executável durante seus estágios de desenvolvimento (como solicitação de proposta, definição de requisitos, especificação de design ou código-fonte). As métricas internas fornecem aos usuários a capacidade de medir a qualidade das entregas intermediárias e, assim, prever a qualidade do produto final. Isso permite que o usuário identifique problemas de qualidade e inicie uma ação corretiva o mais cedo possível no ciclo de vida do desenvolvimento.
- *Externa:* As métricas externas podem ser usadas para medir a qualidade do produto de software, medindo o comportamento do sistema do qual ele faz parte. As métricas externas podem ser usadas apenas durante as etapas de teste do processo do ciclo de vida e durante quaisquer etapas operacionais. A medição é realizada ao executar o produto de software no ambiente do sistema em que ele se destina a operar.

Métricas Internas de Analisabilidade

Métricas internas de analisabilidade indicam uma série de características para prever o esforço ou os recursos gastos do mantenedor ou do usuário para tentar diagnosticar deficiências ou causas de falha, ou até mesmo para identificação de partes que precisam ser alteradas no produto de software. São elas:

- Gravação de atividade:
 - Propósito da métrica: Quão detalhada é a gravação do status do sistema.
 - Método de aplicação: Conte o número de itens registrados no log de atividades conforme especificado e compare-o com o número de itens que precisam ser registrados.
 - Público alvo: Mantenedores e usuários
- Prontidão da função de diagnóstico:
 - Propósito da métrica: Quão abrangente é o fornecimento das funções de diagnóstico.



- Método de aplicação: Conte o número de funções de diagnóstico implementadas conforme especificado e compare-o com o número de funções de diagnóstico necessárias nas especificações.
- Público alvo: Mantenedores e usuários

Métricas Externas de Analisabilidade

Uma métrica de analisabilidade externa deve poder medir atributos como esforço do mantenedor ou usuário ou gasto de recursos ao tentar diagnosticar deficiências ou causas de falhas ou para identificar trechos a serem modificadas. São elas:

- Capacidade de trilha de auditoria
 - Propósito da métrica: O usuário pode identificar uma operação específica que causou falha? O mantenedor pode encontrar facilmente operações específicas que causaram falhas?
 - Método de aplicação: Observar o comportamento do usuário ou mantenedor que está tentando resolver falhas.
 - Público alvo: Desenvolvedor, mantenedor, operador.
- Suporte à função de diagnóstico
 - Propósito da métrica: Qual é a capacidade das funções de diagnóstico no suporte à análise causal? O usuário pode identificar a operação específica que causou a falha? (O usuário pode ser capaz de evitar cair na mesma ocorrência de falha novamente com operação alternativa.) O mantenedor pode encontrar facilmente a causa da falha?
 - Método de aplicação: Observar o comportamento do usuário ou mantenedor que está tentando resolver falhas usando as funções de diagnóstico.
 - Público alvo: Desenvolvedor, mantenedor, operador.
- Capacidade de análise de falhas
 - Propósito da métrica: O usuário pode identificar uma operação específica que causou falha? O mantenedor pode encontrar facilmente a causa da falha?



- Método de aplicação: Observar o comportamento do usuário ou mantenedor que está tentando resolver falhas.
- Público alvo: Desenvolvedor, mantenedor, operador, usuário.
- Eficiência na análise de falhas
 - Propósito da métrica: O usuário pode analisar com eficiência a causa da falha? (O usuário às vezes realiza manutenção definindo parâmetros.) O mantenedor pode encontrar facilmente a causa da falha? Quão fácil é analisar a causa da falha?
 - Método de aplicação: Observar o comportamento do usuário ou mantenedor que está tentando resolver falhas.
 - Público alvo: Desenvolvedor, mantenedor, operador.
- Capacidade de monitoramento de status
 - Propósito da métrica: O usuário pode identificar uma operação específica que causou falha ao obter dados monitorados durante a operação? O mantenedor pode encontrar facilmente a causa da falha obtendo dados monitorados durante a operação?
 - Método de aplicação: Observar o comportamento do usuário ou mantenedor que está tentando obter o status de registro de dados monitorado do software durante a operação.
 - Público alvo: Desenvolvedor, mantenedor, operador, usuário.

Testes usados para verificação e validação

A Analisabilidade tem uma conexão proporcional à complexidade do software em questão. Isto é, em um software em que a complexidade é bem elevada, a capacidade de se analisá-lo em busca de possíveis defeitos se torna bastante difícil.

De acordo com Suhel Ahmad Khana e Raees Ahmad Khana, a Analisabilidade de um código orientado a objetos parte de uma análise numérica de alguns pontos envolvidos neste paradigma de programação como: encapsulamento, herança, *coupling* e especialização. Onde o encapsulamento se refere aos níveis de acesso aos métodos dos objetos; a herança se refere à complexidade envolvida na herança das classes; *coupling* se refere à quantidade de



alterações que uma classe proporciona no comportamento de outra; e especialização se refere ao quão independente uma classe é das demais classes.

Com estes dados recolhidos, faz-se um cruzamento dos mesmos com pesos pré estabelecidos pela pesquisa de Suhel Ahmad Khana e Raees Ahmad Khana. Em seguida, é feita uma análise verificando o desvio dos valores encontrados com os desejados.

Don Coleman and Dan Ash[10] desenvolveram um modelo para a HP Corporate chamado HPMAS: Um modelo hierárquico de avaliação multidimensional. Eles utilizaram um método chamado análise de peso e faixa de pontos de disparo para quantificar a manutenibilidade para obter um “grau de ajuste” e através disso calcular a dimensão da manutenibilidade e calibraram o HPMAS com 16 sistemas de software. O Index da manutenibilidade do HPMAS vai de 0 a 100 em que 100 significa uma excelente manutenibilidade. Eles chegaram a um modelo que utiliza quatro métricas: métrica do esforço de Halstead, métricas com complexidade ciclomática estendida linhas de código e número de comentários. Esse modelo consegue medir a manutenibilidade do produto de software de forma satisfatória.

$$\begin{aligned} \text{Maintainability} &= 171 \\ &- 3.42 \times \ln(\text{ave}E) \\ &- 0.23 \times \text{ave}V(g') \\ &- 16.2 \times \ln(\text{ave}LOC) + \text{ave}CM \end{aligned}$$

Em que aveVol , $\text{ave}V(g')$, $\text{ave}LOC$ e $\text{per}CM$ que respectivamente são , o esforço médio, estendido V (G), linhas médias de código e número de comentários por submódulo (função ou procedimento) no sistema de software.

Eles então aplicaram o modelo a um software em produção e fizeram uma análise pre/post das mudanças na manutenibilidade após o software receber um upgrade e os resultados mostraram o quanto as modificações impactaram a manutenibilidade. Toda vez que o software é modificado ele deve ser passado pelo HPMAS para garantir que a qualidade da manutenibilidade não foi afetada significativamente. Quando ocorrer de a manutenibilidade ser afetada significativamente, isso é um forte indício que a função modificada exigiu muito esforço e representa um problema.



Conclusão

Um modelo de verificação e validação do software é útil pois pode gerar observações mais detalhadas sobre o sistema, representando informações genuínas e economizando então no custo do processo de testes e evolução.

Neste documento, detalhamos o que é Analisabilidade e manutenibilidade para que sejamos capaz de aplicar os conhecimentos no processo de tomada de decisão. Com objetivo de diminuir o desafio de manutenção do software, descrevemos alguns conceitos que diminuem a complexidade total deste processo de testes.

Após examinar alguns modelos aplicados na indústria, concluímos que o propósito de um bom modelo de analisabilidade e manutenibilidade permite com que os envolvidos no projeto possam direcionar seus esforços, pois fornece um feedback mensurável e bastante necessário para os desenvolvedores.



Referências

1. R. Pressman, “Software Engg: A Practitioner’s Approach” , Sixth Ed. Mcgraw-Hill,2005;
2. H. Zuse, “ A Framework of Software Measurement” Walter de Gruyter,1998
3. ISO/IEC 9126-1: 2001 Software Engg.-Product Quality-Part-1: “Quality Model”,Geneva, Switzerland,2001
4. ISO/IEC 9126-4:2004, “Software Engg.: Product Quality-Quality in Use Metrics”, ISO/IEC 2004
5. J. Rumbaugh, M. Blaha, “Object Oriented Modeling and Design”, ISBN:81-7808-738-3, Pearson Education, 2003
- 6.AGUAYO, M.; GUERRA, A.; COLOMBO, R. Avaliação da Manutenibilidade de Produtos de Software. São Paulo, 2005, p. 432-436.
7. GARVIN, David A. "What Does 'Product Quality' Really Mean?" MIT Sloan Management Review, no. 1 (fall 1984), p.25-26.
8. ISO/IEC 9126-4: 2000. Software engineering– Software product quality- Part 4: Quality in use metrics
9. Khana. Suhel, Khana. Raees, “Analyzability Quantification Model of Object Oriented Design”
10. D. Coleman, D. Ash, B. Lowther, and P. Oman. Using metrics to evaluate software system maintainability. Computer, 27(8):44–49, 2002.
11. ISO/IEC 9126-3:2004, “Software Engg.: Product Quality - Part 3: Internal Metrics”, ISO/IEC 2004
12. ISO/IEC 9126-3:2004, “Software Engg.: Product Quality - Part 2: External Metrics”, ISO/IEC 2004