



Estimativa de confiabilidade de sistemas multi-estado k -em- n aplicada a sistemas de informação

Milton Rucks

milton.rucks@ufrgs.br

Flávio S. Fogliatto

Programa de Pós-Graduação em Engenharia de Produção – PPGE/UFGRS

ffogliatto@producao.ufrgs.br

ABSTRACT

Failures representing impacts of different severity may occur in information systems. Reliability analysis models are usually based on the occurrence of failures through the analysis of failure times; such models rarely take into account the severity of failures in the analysis. This paper proposes a model for the reliability analysis of information systems, which are treated as multi-state k -out-of- n systems. The model considers the severity of failures and their impact on reliability figures, providing a better representation of information systems in practice. A case example illustrates the developments proposed in the paper.

Keywords: Software reliability, k -in- n systems, Multistate systems.

RESUMO

Sistemas de informação podem apresentar falhas que representam impactos de diferente severidade para as empresas. Os modelos de análise de confiabilidade usualmente baseiam-se na ocorrência de falhas através da análise de tempos até falha; tais modelos, porém, não contemplam a severidade dos impactos causados pelas falhas. Neste artigo propõe-se um modelo de análise de confiabilidade de sistemas de informação baseado na análise de confiabilidade de sistemas multi-estados k -em- n . O modelo contempla a severidade dos impactos causados pelas falhas, além prover uma melhor representação da realidade operacional de sistemas de informação. Um estudo de caso ilustra os desenvolvimentos propostos no artigo.

Palavras-chave: Confiabilidade de software, Sistemas k -em- n , Sistemas multi-estado.

1 Introdução

Sistemas de *software* são empregados em grande parte das atividades da nossa sociedade. À medida que esses sistemas tornam-se mais presentes, passam a desempenhar funções mais complexas e interagem entre si para compor sistemas maiores. Por consequência, suas funções tornam-se mais críticas, aumentando a relevância de sua disponibilidade operacional. Sistemas de informação como *ERP* (*Enterprise Resource Planning*) e sistemas comerciais são exemplos de sistemas com essas características. Segundo Guerra (2004), esses sistemas podem ser chamados de sistemas modernos críticos, que exigem atenção em relação a sua confiabilidade.

O interesse em determinar a confiabilidade de *software* teve crescimento significativo após o início da década de 1970. De acordo com Souza (2004), até o final da década de 1960 a maior preocupação no desenvolvimento de sistemas era em relação ao desempenho do

hardware, que na época era mais caro que o *software* e mais problemáticos. Os programas computacionais de então eram mais simples e menos suscetíveis a falhas, por apresentarem poucas linhas de código. Após o início da década de 1970, os custos relacionados a *software* passaram a aumentar em relação ao *hardware*, que por sua vez progressivamente se tornou mais barato e confiável.

Existem diversos métodos para a estimativa de confiabilidade de *software* (PELED, 2001). Os métodos mais utilizados são os tradicionais que, de acordo com Cabrera (2006), são funções que combinam o número de defeitos de um *software* e a probabilidade desses defeitos serem acionados. Na verdade, os métodos são todos similares, com pequenas variações (SOUZA, 2004), e não distinguem os tipos de falhas em suas análises. Por outro lado, existem modelos matemáticos

de análise de confiabilidade que são comumente usados em outras áreas da engenharia, como o modelo de sistema k -em- n binário e multi-estado. Segundo Huang e Zuo (2000), modelos de sistemas k -em- n multi-estado são ferramentas de análise mais flexíveis que modelos de sistemas k -em- n binários, e conseguem representar melhor sistemas da área de engenharia.

Levando em consideração que diferentes tipos de falhas em sistemas de informação possuem impactos distintos sobre os usuários e para a empresa a qual servem, se faz necessário um método para a análise de confiabilidade de sistemas que contemple a severidade e abrangência desses impactos. O objetivo do presente artigo é apresentar um método para a estimativa de confiabilidade de sistemas de informação contemplando a severidade dos impactos das falhas. Para tanto, será desenvolvido um método para análise da confiabilidade dos subsistemas que compõem um sistema de informação e, com base nos resultados obtidos, será aplicado um método já existente para determinação da confiabilidade do sistema. O estudo se restringe a análise de falhas dos códigos de *softwares* que compõem o sistema de informação. O método terá como princípio a análise do sistema em sua etapa de funcionamento e técnicas de análise de confiabilidade utilizadas em Engenharia de Confiabilidade.

O artigo contribui de duas formas para o estado da arte sobre confiabilidade de *software*. Primeiro, ao propor um método de análise da confiabilidade de sistemas de gestão, como os ERP, que considere a severidade das falhas nos cálculos. Segundo, ao ilustrar os desenvolvimentos com uma aplicação real, preenchendo uma lacuna por estudos de caso desta natureza na literatura.

O restante deste artigo está assim organizado. A seção 2 destina-se à apresentação do referencial teórico. Nela são apresentadas definições necessárias para a compreensão do trabalho, os métodos mais utilizados para a estimativa de confiabilidade de *software* e suas características, e métodos de análise de confiabilidade de sistemas. Na seção 3 são apresentados os procedimentos metodológicos, destacando-se suas macro-etapas. A seção 4 traz a aplicação prática das proposições metodológicas e seus resultados. Por fim, na seção 5 são apresentadas as conclusões.

2 Referencial Teórico

Grande parte dos métodos atuais utilizados para estimar a confiabilidade de *software* foca na fase de teste, e tem por objetivo estimar a confiabilidade futura. Esses métodos estimam a confiabilidade baseados em dados de falhas e de defeitos coletados durante a fase de teste. Isso ajuda os gerentes de projetos de desenvolvimento de *software* a estimarem o tempo, mão-de-obra e custo para a fase

de testes e as sucessivas correções necessárias, até que o *software* atinja um nível aceitável de confiabilidade (EMAD, 2008).

Modelos matemáticos têm sido desenvolvidos e aplicados para estimar a confiabilidade de *software*. El-Aroui e Soler (1996) usaram um modelo bayesiano para projetar a confiabilidade de *software*, partindo do pressuposto de tempos-entre-falhas exponencialmente distribuídos. Pham e Pham (2001) também propuseram um modelo bayesiano para projetar a confiabilidade futura de *software*, com resultados que superaram outros modelos em termos de precisão da estimativa. Pham e Zhang (2003) propuseram um modelo de predição baseado em Processos Não-Homogêneos de Poisson (NHPP – *Non-Homogeneous Poisson Processes*) para a previsão de confiabilidade de *software*. O modelo se mostrou mais preciso em relação a outros modelos baseados em NHPP disponíveis na literatura.

Técnicas baseadas em auto-aprendizagem também têm sido aplicadas para a predição de confiabilidade de *software*. Um modelo baseado em redes neurais foi proposto por Cai *et al.* (2001). O desempenho do modelo foi avaliado em diferentes arquiteturas de rede e a conclusão foi que o modelo baseado em redes neurais é melhor para a maioria dos casos. Souza (2004) chegou à conclusão de que quando todas as suposições de um modelo analítico são válidas para um projeto específico, o modelo analítico se ajustará melhor aos dados. Porém, para todos os casos em que tais suposições não correspondem à realidade ou não são conhecidas, os modelos de redes neurais apresentam melhor desempenho. Costa *et al.* (2005) usaram programação genética para estimar incrementos na confiabilidade. Foram conduzidos dois experimentos, um baseado no tempo e outro baseado em teste de convergência. Os resultados obtidos nos experimentos atingiram níveis semelhantes a outros modelos tradicionais não-paramétricos. Chen (2007) aplicou *support vector regression* (regressão por vetor de apoio) combinada com algoritmos genéticos para obter a previsão de confiabilidade de um sistema. O resultado superou os modelos de redes neurais e vários outros modelos em relação ao erro quadrado da estimativa.

Segundo Li *et al.* (2005), diferentemente do *hardware*, o *software* não sofre depreciação e sua confiabilidade permanece constante ao longo do tempo, se não houver mudança em seu código. Sempre que houver mudança, entretanto, existe a possibilidade de que a confiabilidade seja alterada. Os métodos apresentados anteriormente não contemplam uma medição final da confiabilidade do sistema e ignoram o fato que devido à dinamicidade do mundo empresarial, muitas empresas são obrigadas a atualizar e modificar seus *softwares* constantemente, alterando a confiabilidade do sistema. Por exemplo, para fabricar um novo produto é necessário cadastrá-lo em todos os *softwares* que a empresa utiliza, colocar

regras de validação nos *softwares* de engenharia, regras para taxas e impostos no sistema fiscal, validações nos sistemas da fábrica e regras para cálculo de comissão e metas nos sistemas de vendas.

Para solucionar esse problema é necessário um método aplicável aos *softwares* em sua fase de operação. Em alguns casos os sistemas de informação são constituídos por múltiplos subsistemas, o que se assemelha a um sistema série-paralelo. Segundo Zaitseva e Levashenko (2006), existem dois tipos básicos de modelos matemáticos para análise de confiabilidade de sistemas: sistemas binários (*binary systems*) e sistemas multi-estados (MSS – *multi-state systems*). Em um sistema binário somente é permitido ao sistema e seus componentes duas possibilidades de estado: falha completa ou funcionamento completo. Essa abordagem não possibilita descrever muitos casos onde o sistema pode ter mais que dois estados distintos. Segundo Zaitseva *et al.* (2007), em um MSS tanto o sistema quanto os componentes podem apresentar mais de dois estados; por exemplo, falha completa, falha parcial, perda de determinada funcionalidade ou funcionamento perfeito.

Segundo Huang e Zuo (2000), o modelo de sistema binário *k-em-n* (*k* é o número mínimo de componentes, de um total de *n* componentes, que devem estar em funcionamento para o sistema funcionar) é comumente usado em problemas práticos de engenharia. Autores como Huang e Zuo (2000), Zuo e Tian (2006), e Li e Zuo (2008) estenderam o conceito de sistema binário *k-em-n* a sistemas multi-estados.

Em aplicações do modelo de sistema multi-estado *k-em-n*, Huang e Zuo (2000) determinaram a probabilidade de funcionamento de cada estado do sistema a partir da probabilidade de funcionamento de cada componente, para cada estado. Foram usados exemplos ilustrativos de aplicação em um sistema de manufatura e em um sistema de controle de qualidade. Essas aplicações demonstram a flexibilidade do modelo, que é reafirmada por Zuo e Tian (2006).

Para definir o algoritmo de avaliação da confiabilidade de um sistema multi-estado *k-em-n* ponderado de Huang e Zuo (2000), usado mais adiante neste artigo, utiliza-se a seguinte notação:

- n Número de componentes do sistema;
- $M+1$ Número total de estados do sistema ou componentes;
- x_i Estado do componente i , $x_i \in \{0, 1, \dots, M\}$;

- x Vetor dos estados dos componentes
 $x = (x_1, x_2, \dots, x_n)$;
- k_j Número mínimo de componentes no estado j para o sistema estar no estado j ou superior;
- $\emptyset(x)$ Função que representa o estado do sistema, $\emptyset(x) \in \{0, 1, \dots, M\}$;
- P_{ij} Probabilidade do componente i estar no estado j ou superior;
- Q_{ij} Probabilidade do componente i estar em um estado inferior a j ; e
- $R_j(n, k_j)$ Probabilidade do sistema estar no estado j ou superior.

O algoritmo de Huang e Zuo (2000) está baseado nos seguintes pressupostos: (i) todos os componentes são independentes; (ii) os estados são ordenados, ou seja, j é um estado melhor que i se $M \geq j \geq i \geq 0$; e (iii) o sistema em estudo é coerente (não há componentes irrelevantes). Os autores propõem a seguinte definição de sistema multi-estado *k-em-n*: um sistema com n componentes é chamado sistema multi-estado *k-em-n* se $\emptyset(x) \geq j$ ($j=1, 2, \dots, M$) sempre que pelo menos k_j componentes estiverem no estado l ou superior, tal que l seja um valor inteiro no intervalo $j \leq l \leq M$.

O sistema é classificado como crescente ou decrescente em função dos valores k_j . Quando $k_1 \geq k_2 \geq \dots \geq k_M$, o sistema é considerado decrescente. Nesse caso, para o sistema estar em um estado j ou superior, existe um número decrescente de componentes necessários que devem estar no estado j ou superior. Quando $k_1 \leq k_2 \leq \dots \leq k_M$, o sistema é considerado crescente. Neste caso, para o sistema estar em um estado j ou superior, um número maior de componentes é necessário. Em outras palavras, existe um número crescente de componentes necessários que devem estar acima de um determinado estado para o sistema estar em um estado superior. Por fim, se $k_j = k_{j+1} = k_{j+2} = \dots = k_M$, então o sistema poderá ser classificado como crescente ou decrescente.

A equação recursiva para determinar o estado de um sistema multi-estado crescente é:

$$R_j(n, k_j) = P_{n,j} R_j(n-1, k_j-1) + Q_{n,j} R_j(n-1, k_j) \quad (1)$$

Os limites para essa equação são:

$$R_j(a, b) = 0 \quad \text{quando } b > a > 0 \quad (2)$$

$$R_j(a, 0) = 1 \quad \text{quando } a \geq 0 \quad (3)$$

3 Procedimentos Metodológicos

O método apresentado é de natureza aplicada e faz uso de uma abordagem quantitativa para estimar a confiabilidade de um sistema de informação. O objetivo é definido como descritivo e o procedimento é experimental. A aplicação do estudo se restringe somente a falhas causadas por erros de código de *software*; falhas de *hardware*, rede, dados e falhas humanas não são consideradas no estudo.

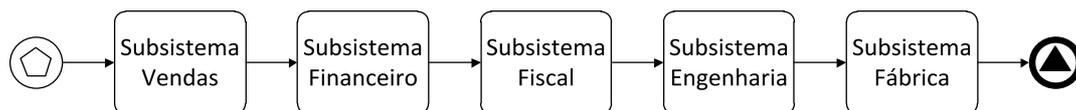


Figura 1 – Fluxo do processamento de pedidos comprados no Brasil

É importante ressaltar que os subsistemas são independentes, ou seja, o desempenho de um subsistema não afeta o desempenho dos demais. Os pedidos são processados pelos subsistemas seguindo uma lógica FIFO (*first in, first out*), porém, de forma análoga aos subsistemas, são processados de forma independente, fazendo com que o travamento de um pedido não afete o processamento dos demais. A abrangência do estudo considera a operação dos subsistemas durante o período de um ano, de 01 janeiro de 2010 a 31 de dezembro de 2010. Esse período foi determinado em função da implementação do subsistema fábrica, que ocorreu em Agosto de 2009 e necessitou três meses para estabilizar a operação.

O modelo matemático usado para o cálculo da confiabilidade do sistema será o proposto por Huang e Zuo (2000), detalhado na seção 2. A escolha desse modelo justifica-se devido a sua capacidade de contemplar os impactos das falhas e a flexibilidade de adaptação a diversos sistemas. Em Huang e Zuo (2000), o modelo foi usado para determinar a confiabilidade de sistemas de manufatura, controle de qualidade e frota de tratores, tendo sido aplicado a diversos sistemas genéricos em Zuo e Tian (2006) e Li e Zuo (2008). A seguir são apresentadas as macro etapas propostas para a aplicação do método, contextualizadas ao estudo de caso.

3.1 Definição de critérios para a classificação dos estados

Para classificar os tipos de falha, define-se que os subsistemas podem operar em quatro estados distintos de funcionamento (Tabela 1)

Tabela 1 – Estados de funcionamento

Estado	Descrição do estado
3	Funcionamento perfeito
2	Apresenta alguma falha de baixo impacto (nível de usuário único)
1	Apresenta alguma falha de médio impacto (impacta uma equipe de trabalho)
0	Apresenta alguma falha de impacto severo (impacta todos os usuários do subsistema)

A empresa abordada no estudo de caso pertence ao ramo de tecnologia. O alto grau de customização e volume exigem um complexo sistema de informação para viabilizar a operação. A empresa possui diversos subsistemas especializados para o processamento dessas informações. Para a operação no Brasil, o fluxo dos pedidos passa pelos subsistemas apresentados na Figura 1.

Para determinar em qual estado de funcionamento o subsistema se encontra, são adotados os seguintes critérios. No estado 3, o subsistema não apresenta nenhuma das falhas citadas nos demais estados. No estado 2, o subsistema apresenta uma ou mais falhas que afetam um único usuário (por exemplo, a impossibilidade temporária de extrair um relatório específico). No estado 1, o subsistema apresenta uma ou mais falhas que impactam um grupo inteiro de trabalho (por exemplo, a impossibilidade de consultar dados de cartão de crédito de um determinado banco). No estado 0, o subsistema apresenta uma ou mais falhas que impactam todos os seus usuários (por exemplo, o sistema de vendas não processa nenhum pedido).

Esses critérios são adotados também em função dos valores internos da empresa. A empresa em estudo possui um sistema para gerenciamento de problemas relacionados a TI, onde o próprio sistema de gerenciamento define um peso e uma prioridade para a resolução do problema, em função do número de usuários afetados (usuário único, grupo de trabalho ou todos os usuários).

3.2 Cálculo da confiabilidade dos subsistemas para cada estado

Os subsistemas em análise podem apresentar mais de uma falha ao mesmo tempo; por exemplo, o subsistema apresenta a falha A para um usuário e a falha B para outro usuário simultaneamente, o que gera uma sobreposição nos tempos de ocorrência dos estados de falha. Além disso, para que o método de Huang e Zuo (2000) seja aplicável, a confiabilidade do subsistema deve ser medida em função do impacto sobre o usuário final. Devido ao fato de os subsistemas possuírem múltiplos usuários e os registros de falhas indicarem falhas de diferentes impactos (falhas que impactam um único usuário, grupo de trabalho, ou todos os usuários), para obter a confiabilidade dos subsistemas é necessário fazer uma ponderação entre o tempo total da missão do

subsistema, o tempo total de funcionamento no estado j , e o número de usuários afetados pela falha [equações (6), (7) e (8)].

Para o cálculo das probabilidades de funcionamento dos subsistemas, define-se a seguinte notação:

f_{ij} Falha do subsistema i que caracteriza o estado j ;

F_{ij} Número total de falhas do subsistema i que caracteriza o estado j ;

$t_{f_{ij}}$ Tempo de duração da falha;

T_{ij} Somatório dos tempos de falha do subsistema i que caracterizam o estado j [equação (4)];

U_i Número total de usuários do subsistema i ;

G_i Número total de grupos de trabalho que utilizam o subsistema i ;

A_i Média de usuários por grupo de trabalho [equação (5)];

m Tempo de duração da missão.

$$T_{ij} = \sum_{f_{ij}} F_{ij} = \uparrow^{t_{f_{ij}}} \quad (4)$$

$$A_i = \frac{U_i}{G_i} \quad (5)$$

Confiabilidade do subsistema i para o estado 0:

$$P_{i,0} = \frac{T_{i,0} \cdot U_i}{m \cdot U_i} \quad (6)$$

Confiabilidade do subsistema i para o estado 1:

$$P_{i,1} = \frac{T_{i,1} \cdot A_i}{m \cdot U_i} \quad (7)$$

Confiabilidade do subsistema i para o estado 2:

$$P_{i,2} = \frac{T_{i,2}}{m \cdot U_i} \quad (8)$$

Confiabilidade do subsistema i para o estado 3:

$$P_{i,3} = w - P_{i,0} - P_{i,1} - P_{i,2} \quad (9)$$

Assim, o somatório dos tempos de falha que apresente impacto a todos os usuários é multiplicado pelo número total de usuários, o somatório dos tempos de falha que apresente impacto a um grupo de trabalho é multiplicado pelo número médio de usuários dos grupos de trabalho e o somatório dos tempos de falha que apresente impacto em um único usuário não possui nenhum multiplicador. Ao final da análise, será possível obter as probabilidades informadas na Tabela 2.

Tabela 2 – Probabilidade de operação dos subsistemas para os estados

Estados	Subsistema					
	Vendas	Financeiro	Fiscal	Engenharia	Fábrica	
	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	
Estado 3	$j=3$	$P_{1,3}$	$P_{2,3}$	$P_{3,3}$	$P_{4,3}$	$P_{5,3}$
Estado 2	$j=2$	$P_{1,2}$	$P_{2,2}$	$P_{3,2}$	$P_{4,2}$	$P_{5,2}$
Estado 1	$j=1$	$P_{1,1}$	$P_{2,1}$	$P_{3,1}$	$P_{4,1}$	$P_{5,1}$
Estado 0	$j=0$	$P_{1,0}$	$P_{2,0}$	$P_{3,0}$	$P_{4,0}$	$P_{5,0}$

3.3 Cálculo da probabilidade de funcionamento do sistema para os estados

Usando o modelo matemático proposto por Huang e Zuo (2000), é possível determinar a probabilidade de funcionamento do sistema como um todo para cada estado e construir a Tabela 3, onde a probabilidade do sistema estar no estado j é dada pela equação (10).

$$Pr(\emptyset = j) = Pr(\emptyset \geq j) - Pr(\emptyset \geq j+1) \quad (10)$$

Tabela 3 – Probabilidade de funcionamento do sistema para os estados 4 APLICAÇÃO E RESULTADOS

Estado	Probabilidade do sistema para cada estado
Estado 3 $j=3$	$Pr(\emptyset \geq 3) = A$ $Pr(\emptyset = 3) = A$
Estado 2 $j=2$	$Pr(\emptyset \geq 2) = B$ $Pr(\emptyset = 2) = F$
Estado 1 $j=1$	$Pr(\emptyset \geq 1) = C$ $Pr(\emptyset = 1) = G$
Estado 0 $j=0$	$Pr(\emptyset \geq 0) = 1$ $Pr(\emptyset = 0) = H$

Caso sejam necessárias alterações, o método irá indicar os subsistemas com confiabilidade mais baixa e aqueles que apresentam maior impacto sobre o sistema como um todo. Por consequência, será possível determinar quais os pontos necessários de intervenção para a otimização da confiabilidade do sistema.

4 Aplicação e Resultados

4.1 Cálculo da confiabilidade dos subsistemas para cada estado

O número total de usuários e o número total de grupos de trabalho de cada subsistema é conhecido, sendo informados na Tabela 4.

Através da aplicação da equação (4) aos dados de relatórios de falhas, é determinado o somatório dos tempos de falha dos subsistemas para cada estado (Tabela 5), medido em horas.

Tabela 4 - G_i, U_i, A_i dos subsistemas

	Subsistema				
	Vendas	Financeiro	Fiscal	Engenharia	Fábrica
	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
G_i	101	16	31	2	14
U_i	1165	193	251	30	297
A_i	11,534	12,062	8,0967	15	21,214

Tabela 5 – Somatório dos tempos de falha dos subsistemas

Estados		Subsistema				
		Vendas	Financeiro	Fiscal	Engenharia	Fábrica
		$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
Estado 2	$j=2$	23801:02:54	1903:09:09	8228:21:05	564:32:32	3356:15:16
Estado 1	$j=1$	5494:15:23	3859:50:53	4465:19:28	66:35:12	1084:27:16
Estado 0	$j=0$	28:08:23	66:50:25	88:06:05	0:00:00	16:13:00

A análise desejada é para uma missão de um ano, o que corresponde a 8760 horas. Aplicando-se as equações (6) a (9), é possível obter as informações na Tabela 6. Como exemplo ilustrativo, são calculadas as probabilidades de funcionamento do subsistema de vendas utilizando as equações (11) a (13).

Tabela 6 – Probabilidades de funcionamento dos subsistemas

Estados		Subsistema				
		Vendas	Financeiro	Fiscal	Engenharia	Fábrica
		$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
Estado 3	$j=3$	0,988245618	0,963705288	0,969757261	0,99405121	0,988016185
Estado 2	$j=2$	0,002332201	0,001125673	0,003742269	0,00214818	0,001290014
Estado 1	$j=1$	0,006209883	0,027538870	0,016443233	0,00380061	0,008842584
Estado 0	$j=0$	0,003212297	0,007630169	0,010057236	0	0,001851218

$$P_{1,0} = \frac{T_{1,0} \cdot U_1}{m \cdot U_1} = \frac{28.139 \cdot 1165}{8760 \cdot 1165} = 0,003212297 \tag{11}$$

$$P_{1,1} = \frac{T_{1,1} \cdot A_1}{m \cdot U_1} = \frac{5494.256 \cdot 11.53465}{8760 \cdot 1165} = 0,006209883 \tag{12}$$

$$P_{1,2} = \frac{T_{1,2}}{m \cdot U_1} = \frac{23801.048}{8760 \cdot 1165} = 0,002332201 \tag{13}$$

$$P_{1,3} = 1 - 0,003212297 - 0,006209883 - 0,002332201 = 0,988245618 \tag{14}$$

4.2 Aplicação do método de Huang e Zuo

O sistema k -em- n analisado se caracteriza por ser um sistema em série, ou seja, um sistema 5-em-5 para todos os estados, o que significa que todos os subsistemas devem estar funcionando no estado j para que o sistema esteja no estado j . Assim, será necessária a obtenção dos termos $R_3(5,5), R_2(5,5), R_1(5,5)$ e $R_0(5,5)$.

Aplicando-se a equação (1), as condições de contorno (2) e (3), e posteriormente fazendo uma substituição recursiva, é possível obter a probabilidade de funcionamento do sistema para cada estado [Tabelas 7 a 10].

Tabela 7 – Cálculo do termo $R_3(5,5)$

Termo	Cálculo do termo	Resultado
$R_3(5,5)$	$P_{5,3} R_3(4,4) + Q_{5,3} R_3(4,5) = P_{5,3} R_3(4,4)$	0,9070788
$R_3(4,4)$	$P_{4,3} R_3(3,3) + Q_{4,3} R_3(3,4) = P_{4,3} R_3(3,3)$	0,9180809
$R_3(3,3)$	$P_{3,3} R_3(2,2) + Q_{3,3} R_3(2,3) = P_{3,3} R_3(2,2)$	0,9235750
$R_3(2,2)$	$P_{2,3} R_3(1,1) + Q_{2,3} R_3(1,2) = P_{2,3} R_3(1,1)$	0,9523775
$R_3(1,1)$	$P_{1,3} R_3(0,0) + Q_{1,3} R_3(0,1) = P_{1,3} R_3(0,0)$	0,9882456
$R_3(1,1)$	$R_3(0,0)$	1

Tabela 8 – Cálculo do termo $R_2(5,5)$

Termo	Cálculo do termo	Resultado
$R_2(5,5)$	$P_{5,2} R_2(4,4) + Q_{5,2} R_2(4,5) = P_{5,2} R_2(4,4)$	0,9169646
$R_2(4,4)$	$P_{4,2} R_2(3,3) + Q_{4,2} R_2(3,4) = P_{4,2} R_2(3,3)$	0,9268765
$R_2(3,3)$	$P_{3,2} R_2(2,2) + Q_{3,2} R_2(2,3) = P_{3,2} R_2(2,2)$	0,9304126
$R_2(2,2)$	$P_{2,2} R_2(1,1) + Q_{2,2} R_2(1,2) = P_{2,2} R_2(1,1)$	0,9557401
$R_2(1,1)$	$P_{1,2} R_2(0,0) + Q_{1,2} R_2(0,1) = P_{1,2} R_2(0,0)$	0,9905778
$R_2(1,1)$	$R_2(0,0)$	1

Tabela 9 – Cálculo do termo $R_1(5,5)$

Termo	Cálculo do termo	Resultado
$R_1(5,5)$	$P_{5,1} R_1(4,4) + Q_{5,1} R_1(4,5) = P_{5,1} R_1(4,4)$	0,9774208
$R_1(4,4)$	$P_{4,1} R_1(3,3) + Q_{4,1} R_1(3,4) = P_{4,1} R_1(3,3)$	0,9792336
$R_1(3,3)$	$P_{3,1} R_1(2,2) + Q_{3,1} R_1(2,3) = P_{3,1} R_1(2,2)$	0,9792336
$R_1(2,2)$	$P_{2,1} R_1(1,1) + Q_{2,1} R_1(1,2) = P_{2,1} R_1(1,1)$	0,9891820
$R_1(1,1)$	$P_{1,1} R_1(0,0) + Q_{1,1} R_1(0,1) = P_{1,1} R_1(0,0)$	0,9967877
$R_1(1,1)$	$R_1(0,0)$	1

Tabela 10 – Cálculo do termo $R_0(5,5)$

Termo	Cálculo do termo	Resultado
$R_0(5,5)$	$P_{5,0} R_0(4,4) + Q_{5,0} R_0(4,5) = P_{5,0} R_0(4,4)$	1
$R_0(4,4)$	$P_{4,0} R_0(3,3) + Q_{4,0} R_0(3,4) = P_{4,0} R_0(3,3)$	1
$R_0(3,3)$	$P_{3,0} R_0(2,2) + Q_{3,0} R_0(2,3) = P_{3,0} R_0(2,2)$	1
$R_0(2,2)$	$P_{2,0} R_0(1,1) + Q_{2,0} R_0(1,2) = P_{2,0} R_0(1,1)$	1
$R_0(1,1)$	$P_{1,0} R_0(0,0) + Q_{1,0} R_0(0,1) = P_{1,0} R_0(0,0)$	1
$R_0(0,0)$	$R_0(0,0)$	1

Aplicando a equação (10), é possível calcular a probabilidade do sistema para cada estado [equações (15) a (17)], obtendo como resultado final as probabilidades de funcionamento do sistema, apresentadas na Tabela 11.

$$\Pr\emptyset = 2 = \Pr\emptyset \geq 2 - \Pr\emptyset \geq 3 = 0,9169646 - 0,9070788 = 0,0098559 \quad (15)$$

$$\Pr\emptyset = 1 = \Pr\emptyset \geq 1 - \Pr\emptyset \geq 2 = 0,9774208 - 0,9169646 = 0,0639545 \quad (16)$$

$$\Pr\emptyset = 0 = \Pr\emptyset \geq 0 - \Pr\emptyset \geq 1 = 1 - 0,9774208 = 0,0225792 \quad (17)$$

Tabela 11 – Probabilidade de funcionamento do sistema

Estado	Probabilidade do sistema para cada estado	
Estado 3	$\Pr(\emptyset \geq 3) = 0,9070788$	$\Pr(\emptyset = 3) = 0,9070788$
Estado 2	$\Pr(\emptyset \geq 2) = 0,9169646$	$\Pr(\emptyset = 2) = 0,0098559$
Estado 1	$\Pr(\emptyset \geq 1) = 0,9774208$	$\Pr(\emptyset = 1) = 0,0639545$
Estado 0	$\Pr(\emptyset \geq 0) = 1$	$\Pr(\emptyset = 0) = 0,0225792$

4.3 Resultados

Através da análise da Tabela 6, é possível constatar que os subsistemas financeiro e fiscal apresentam os piores índices de funcionamento em perfeito estado ($P_{2,3}=0,963705288$ e $P_{3,3}=0,969757261$, respectivamente). Além disso, possuem as duas maiores probabilidades globais de estado de falha ($P_{2,1}=0,027538870$ e $P_{3,1}=0,016443233$).

Para o sistema, a alta probabilidade de ocorrência do estado 1 (0,0639545) se justifica devido ao fato de existir uma grande chance de um problema impactar um grupo inteiro de trabalho, já que é característica da empresa vários funcionários executarem a mesma função dentro de um mesmo grupo.

É importante ressaltar que nem sempre uma falha apresenta impactos diretos no processamento dos pedidos. Existem diversos tipos de falhas (por exemplo, lentidão para a extração de relatórios de um subsistema) que podem impactar todos os usuários do subsistema, porém não apresentam impacto algum no processamento dos pedidos.

É difícil traduzir em termos gerenciais o significado da confiabilidade do estado de funcionamento perfeito do sistema (0,9070788). A escassez de literatura sobre o assunto e o método adotado não permitem comparar esse resultado a aplicações em sistemas semelhantes. Além de existir a possibilidade das falhas não apresentarem impactos diretos no processamento dos pedidos, esse número de confiabilidade não faz nenhuma ponderação do grau de importância dos subsistemas. Por exemplo, uma falha de estado 0 que paralise o subsistema vendas não permitirá a realização de nenhuma venda, gerando perda de clientes e um custo de imagem negativa junto aos clientes que tentam comprar por telefone ou através da internet, enquanto que o transtorno causado por uma paralisação total no subsistema financeiro pode ser contornado com horas extras de trabalho dos funcionários atingidos pela falha. Os valores da confiabilidade dos subsistemas podem fornecer informações gerenciais mais úteis; por exemplo, indicar qual subsistema apresenta os melhores e piores índices de confiabilidade, e com base nisso direcionar recursos para projetos de melhorias.

Devido à propriedade multiplicativa do método, para otimizar a confiabilidade do sistema no estado de perfeito funcionamento é necessário que todos os subsistemas apresentem o maior valor possível de confiabilidade

e a menor diferença possível da média. Os subsistemas fiscal e financeiro são os únicos que se encontram abaixo da média (Tabela 12), o que permite concluir que é

necessário prioritariamente aumentar a confiabilidade desses subsistemas para otimizar a confiabilidade global do sistema.

Tabela 12 – Diferença da média de probabilidade do estado 3

	Subsistema				
	Vendas	Financeiro	Fiscal	Engenharia	Fábrica
	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
Estado 3 (j=3)	0,988245618	0,963705288	0,969757261	0,99405121	0,988016185
Diferença da média	0,007490506	-0,01704982	-0,01099785	0,013296098	0,007261073
Média	0,980755112				

5 Conclusão

Neste artigo apresentou-se um modelo matemático de análise de confiabilidade de sistema multi-estado para determinar a confiabilidade de *softwares* de sistemas de informação. O objetivo foi desenvolver um método para a estimativa de confiabilidade de sistemas de informação que contemplasse a severidade dos impactos das falhas para a empresa e seus usuários, estando restrito à análise de falhas do código dos *softwares* que compõem o sistema.

Para solucionar o problema foi usado o modelo matemático proposto por Huang e Zuo (2000). O modelo traz como resultado final a confiabilidade do sistema de informação, levando em consideração o grau de impacto das falhas, que são representadas pelo modelo como os estados de funcionamento. Para isso foi necessário estabelecer os critérios que determinam os estados dos subsistemas, e desenvolver um método para determinar a confiabilidade de cada subsistema em função desses estados e, posteriormente, equiparar os subsistemas a componentes de um sistema.

Como principal resultado obtido, foi determinada a confiabilidade do sistema de informação dividida em seus estados de funcionamento. A confiabilidade do sistema é de 0,9070788 para o estado de perfeito funcionamento, para uma missão de um ano de duração. Também foi possível constatar que os subsistemas financeiro e fiscal apresentam os piores índices de funcionamento no estado perfeito, o que possui grande impacto no sistema como um todo.

É indicado para estudos futuros realizar uma análise complementar relacionando a confiabilidade encontrada a valores financeiros dos pedidos que são processados pelo sistema. Ao relacionar a confiabilidade a valores financeiros, é possível comparar o resultado ao custo de melhoria da confiabilidade e, baseado nessas conclusões, tomar ações efetivas para otimizar a confiabilidade do sistema em função do custo.

6 Referências

- CABRERA, E.V. *Software reliability methods*. Technical Report. Technical University of Madrid, Aug, 2006.
- CAI, K.; CAI, L. WANG, W.; YU, Z.; ZHANG, D. *On the neural network approach in software reliability modeling*. The Journal of Systems and Software, v. 58, n. 1, p. 47-62, Aug. 2001.
- CHEN, K. *Forecasting systems reliability based on support vector regression with genetic algorithms*. Reliability Engineering and System Safety, v. 92, n.4, p. 423-432, Apr. 2007.
- COSTA, E.; VERGILIO, S.; POZO, A.; SOUZA, G. *Modeling software reliability growth with genetic programming*. 16th IEEE international symposium on software reliability engineering, p. 180-190, Nov. 2005
- EMAD, A. E. *Software reliability identification using functional networks: A comparative study*, Journal Elsevier Expert Systems with Applications, v. 36, n. 2, p. 4013-4020, Mar. 2008
- EL-AROU, M.; SOLER, J. *A bayes nonparametric framework for software reliability analysis*. IEEE Transactions on Reliability, v. 45, n. 4, p. 652-660, Dec.1996.
- GUERRA, P. A. C. *Uma abordagem arquitetural para tolerância a falhas em sistemas de software baseados em componentes*. Tese de Doutorado, IC, Unicamp, 2004.
- HUANG, J.; ZUO, M. *Multi-state k-out-of-n system model and its applications*. IEEE Proceedings Annual Reliability and Maintainability Symposium, p. 264-268, Aug. 2000.
- LI, R.; YU, T.; FANG, M. *Reliability management for information system*. Journal of Shanghai University, v. 9, n. 3, p. 268-274, Jun. 2005.

LI, W.; ZUO, M. *Reliability evaluation of multi-state weighted k-out-of-n systems*. Reliability Engineering & System Safety, v. 93, n.1, p. 160-167, Jan. 2008.

PELED, D.A. *Software Reliability Methods*. Springer, USA: Springer, 2001. 331p.

PHAM, L.; PHAM, H. *A bayesian predictive software reliability model with pseudo-failures*. IEEE Transactions on Systems, v. 31, n. 3, p. 233-238, Aug. 2001.

PHAM, H.; ZHANG, X. *NHPP software reliability and cost models with testing coverage*. European Journal of Operational Research. v. 145, n. 2, p. 443-454, Mar. 2003.

SOUZA, G. *Utilizando redes neurais artificiais para modelar a confiabilidade de software*. Tese de mestrado. Departamento de Informática – Universidade Federal do Paraná (UFPR), 2004.

ZAITSEVA, E.; LEVASHENKO, V. *Dynamic reliability indices for parallel, series and k-out-of-n multi-state system*. IEEE Annual Reliability and Maintainability Symposium, p. 253-259, Jan. 2006.

ZAITSEVA, E.; LEVASHENKO, V.; MATIASKO, K. *Dynamic reliability indices for series and parallel multi-state system*. IEEE The international Conference on “Computer as a Tool”, p. 511-518, Sep. 2007.

ZUO, M.; TIAN, Z. *Performance evaluation of generalized multi-state k-out-of-n systems*. IEEE Transactions on reliability, v. 5, n. 2, p. 319 - 327, Jun. 2006.

Recebido em: 21/05/2011.

Aceito em: 17/08/2011