

Qualidade em Sistemas de Informações baseada em Requisitos Não Funcionais

José Gladistone Rocha

Resumo— O objetivo desse trabalho é apresentar uma lista base dos requisitos não funcionais e seus índices a serem necessariamente considerados no desenvolvimento de um sistema de TI. Projetos de desenvolvimento de sistemas de Tecnologia da Informação (TI) podem fracassar devido à falta de observância rigorosa de requisitos não funcionais, como o não atendimento ao ambiente de infraestrutura de implantação, a perda de performance do sistema, e a falta de portabilidade para outras plataformas. Neste contexto, Será disponibilizada uma discussão sobre o impacto desses índices na qualidade do sistema final considerando especialmente a fase de elicitação de requisitos. Para tanto, foi realizada uma compilação de trabalhos científicos e de livros consagrados na área de Engenharia de *Software*, além da realização de um estudo de caso real. Os resultados finais permitem conjecturar sobre a significativa mitigação dos riscos na entrega dos produtos de *software* aos usuários.

Palavras-chave—Qualidade; Qualidade de *Software*; Requisito Não funcional; Engenharia de *Software*.

Abreviações—Engenharia de *Software* (ES); Requisito Funcional (RF); Requisito Não Funcional (RNF); Sistema de Informações (SI); Tecnologia da Informação (TI); Engenharia de Requisitos (ER)

1 Introdução

O desenvolvimento de um sistema de TI consiste na construção de um produto de *software* que é fruto de um trabalho de Engenharia de *Software* (ES), com aplicação de métodos e técnicas para garantir resultados com qualidade, sem comprometer orçamento e prazo, bem como prover a satisfação final do cliente (SOMMERVILLE, 2011)(ROCHA e RODRIGUES, 2015).

A deficiência de qualidade é um dos fatores que conduzem ao fracasso os sistemas de TI (SOMMERVILLE, 2011). Ao entrar em produção, tornam-se pouco robustos ou fragilizados quanto a sua arquitetura. Isso, muitas vezes, é criticado pelos usuários. Tal deficiência se torna um problema, pois muito esforço é despendido ao longo do projeto de desenvolvimento, principalmente relacionado a questões

financeiras, e descontentamento por parte dos *stakeholders* do projeto de TI (BEZERRA, 2007).

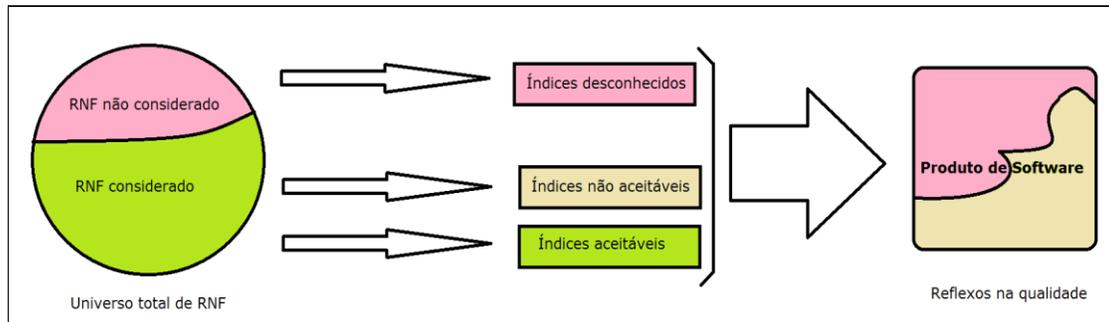


Figura 1: Universo de RNF e seus reflexos na qualidade de *software*.

Fonte: os autores

Sistemas de TI apresentam deficiências na sua qualidade simplesmente porque seus desenvolvedores não atentam rigorosamente aos critérios baseados nos Requisitos não Funcionais (RNF) (SOMMERVILLE, 2011). A correta e minuciosa especificação de RNF é determinante para a qualidade final do *software* (ALSHAZLY; ELFATATRY; ABOUGABAL, 2014) apud (WOHLIN, 2005), a qual inclui a robustez do sistema e o grau de satisfação do usuário final. Esse processo de captura e elicitação de RNF pode ser esquematizado como indicado na Figura 1.

Como se observa na Figura 1, muitos dos RNF considerados para os sistemas de TI não possuem índices aceitáveis, já para outros RNF considerados possuem índices desconhecidos. Isso reflete aos sistemas de TI quando entram em produção, muitas vezes, em custos elevados na fase de manutenção desses sistemas.

Neste contexto, o objetivo deste trabalho é apresentar uma lista, a mais completa possível, dos RNFs e seus índices associados em forma de *checklist*, a qual deverá ser levada em consideração na construção de sistemas de TI. Isso auxiliará equipes desenvolvedoras nas fases iniciais de desenvolvimento, particularmente na elicitação de requisitos.

O restante deste texto é organizado como descrito a seguir. A Seção 2 são apresentados os termos utilizados no trabalho. A Seção 3 traz conceitos e definições que são utilizadas ao longo deste texto para facilitar o entendimento do leitor. A Seção 4

discorre sobre os RNFs e seus índices. A Seção 5 trata de um estudo de caso real. E a Seção 6 trata das conclusões e de trabalhos futuros.

2 Conceitos e definições

Para facilidade de compreensão do conteúdo deste artigo, são aqui adotados os seguintes conceitos e definições existentes na literatura:

a) Atributo: Propriedade mensurável, física ou abstrata, de uma entidade (BRASIL, 2001).

b) Requisito: é uma condição ou uma capacidade com a qual o sistema deve estar de acordo (RUP, 2007);

c) Elicitação de Requisitos: corresponde ao processo de levantamento de requisitos (LAUESEN, 2002) apud (BARBOSA et al, 2009);

e) Requisito Não Funcional: trata-se de uma restrição aos serviços ou funções oferecidos pelo sistema, onde tal restrição refere-se ao processo de desenvolvimento e/ou imposição pelas normas. Muitas vezes se aplica ao sistema como um todo (SOMMERVILLE, 2011). Por sua vez, Dennis, Wixom e Roth *apud International Institute of Business Analysis (IIBA)* (2014) define Requisito Não Funcional como sendo “os atributos de qualidade, as restrições de design e implementação e as interfaces externas que um produto deve possuir”;

f) Requisito de qualidade interna: especificam o nível de qualidade requerido sob o ponto de vista interno do produto. Tais requisitos são usados para especificar as propriedades dos produtos intermediários. Estes podem incluir modelos estáticos e dinâmicos, outros documentos e código-fonte. Podem ser usados como metas para validação em vários estágios de desenvolvimento. Eles também podem ser usados para definir estratégias de desenvolvimento e critérios de avaliação e de verificação durante o desenvolvimento. Isto pode incluir o uso de métricas adicionais, por exemplo, reusabilidade. Convém que esses requisitos sejam definidos quantitativamente usando métricas internas (BRASIL, 2003);

g) Sistema: é um conjunto de elementos que interagem para realizar objetivos (STAIR, REYNOLDS, 2011, p8);

h) Sistema de Informações: é um conjunto de elementos ou componentes inter-relacionados que coleta, manipula, armazena e disseminam dados e informações, e fornece uma reação coletiva para alcançar um objetivo (STAIR, REYNOLDS, 2011, p8);

i) Stakeholder: é uma pessoa ou papel que, de alguma maneira, é afetado pelo sistema e têm necessidades diferentes e muitas vezes inconsistentes entre si (SOMMERVILLE, 2011);

j) Produto de Software: conjunto de programas de computador, procedimentos e possível documentação e dados associados (BRASIL, 2001);

k) Tecnologia de Informação (TI): é o conjunto de todas as atividades e soluções providas por recursos de computação que visam a produção (Wikipédia, 2017).

l) Sistema de Informação (SI): é um conjunto de componentes inter-relacionados que coleta, manipula, armazena e disseminam dados e informações e que ajuda e fornece um mecanismo de realimentação para atingir um objetivo (STAIR E REYNOLDS, 2011)

m) Qualidade: totalidades de características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas (Site TI & Gestão, 2016).

3 Trabalhos relacionados

Shaha e Patel (2016) afirmam que muitos erros podem originar/propagam a partir da fase de requisitos, causada por requisitos mal escritos, ambíguo, pouco claras ou não atendidos. Os autores categoricamente explicitam na falha em identificar requisitos, e como isso ocorre, há de se considerar que provavelmente alguns, para não dizer muitos, RNF deixam de serem considerados e com isso resultando na qualidade de sistemas de TI, foco desse estudo.

Afzal, Torkar e Feld (2008) apontam ainda como RNF: usabilidade, segurança, robustez, capacidade, integridade, eficiência, confiabilidade, manutenibilidade, testabilidade, flexibilidade, reusabilidade, portabilidade, interoperabilidade,

desempenho, disponibilidade e escalabilidade. Todos esse requisitos foram contemplados no *checklist* a ser escriturado neste artigo,

A norma ABNT ISO/IEC 9126-1 (BRASIL, 2003), por sua vez, apresenta as características componentes do modelo de qualidade para a qualidade interna e externa ao produto de *software*, a saber: a) Funcionalidade; b) Confiabilidade; c) Usabilidade; d) Eficiência; e) Manutenibilidade; e f) Portabilidade. Excluída a primeira característica, por não ser objeto desse estudo, as demais serão consideradas no *checklist* desse estudo.

Dennis, Wixom e Roth (2014) consideram como categorias de RNF: operacional, desempenho, segurança, cultural e política. Acrescentam ainda que os RNF não descrevem processos ou informações do negócio, mas são muito úteis para o entendimento de como o sistema final deve ser semelhante.

El-Hadary e El-Kassas (2014) ao abordar em seu trabalho sobre captura de requisitos de segurança em SI definem categorias a serem assistidas como sendo: confidencialidade, integridade, disponibilidade responsabilidade e autenticidade.

Para Sommerville (2011), o universo dos RNF são: a) Tempo de resposta; b) Utilização de memória; c) Desempenho; d) Proteção; e) Robustez; e f) Confiabilidade. O autor afirma que em qualidade deve-se incluir o comportamento do *software* durante sua execução; a estrutura e organização dos programas do sistema; e a documentação associada. Isto se reflete na chamada qualidade ou atributos não funcionais do *software*. Exemplos desses atributos são: tempo de resposta do *software* a uma consulta do usuário e a compreensibilidade do código do programa (SOMMERVILLE, 2011). Ainda retrata que a qualidade de *software*, e conseqüentemente do SI, está associado diretamente com os RNF a ele inerentes. O nível de qualidade atingido se deve pelo atendimento aos índices de aceitação dos RNF.

Pressman (2001) *apud* Olsina (1999) ao reportar qualidade de aplicações *web*, apresenta cinco características que compõem a árvore de requisitos de qualidade a serem perseguidas pelos desenvolvedores de *software* sendo: usabilidade, funcionalidade, confiabilidade, eficiência e manutenibilidade.

Barbosa et al (2009) afirmam que um conjunto de técnicas de elicitação pode ser aplicado para auxiliar analistas e usuários na identificação e definição de requisitos,

com vistas a garantir que os requisitos relevantes de um SI sejam elicitados de forma correta e completa. Isso permite se obter qualidade em *software*. Os autores acrescentam que a atividade de elicitar requisitos não se resume apenas à aplicação de uma técnica, mas também na colaboração entre analistas e clientes.

Por fim, Shah e Patel (2016) definem um esquema para elicitar requisitos não funcionais onde contempla os seguintes tipos e subtipos de RNF: confiabilidade; disponibilidade; segurança, com seus subtipos: autenticação e autorização; privacidade, com seus subtipos: anonimato, desvinculação, indetectabilidade, pseudonimato, gerenciamento de identidade; desempenho; portabilidade; usabilidade; interoperabilidade; e escalabilidade.

4 Requisitos Não Funcionais e seus índices

Os requisitos não funcionais expressam como os RF, depois de implementados, se comportaram na utilização dos serviços ou funcionalidades disponibilizados pelos RF Shah e Patel (2016).

Apesar da evolução da Engenharia de Requisitos (ER) ainda não está madura o suficiente para auxiliar na captura e aderência de RNF aos sistemas construídos (SHAH E PATEL, 2016), (SOMMERVILLE, 2011), (PRESSMAN, 2001) *apud* (OLSINA, 1999), (EL-HADARY e EL-KASSAS, 2014) (Dennis, Wixom e Roth 2014). Isso sugere uma potencial fragilidade nos sistemas desenvolvidos sem dar a devida atenção às características não funcionais a que tais SI devem abranger.

Muito foi realizado no sentido de criar tecnologias para a análise funcional de *software*, já quantos aos RNF não houve a mesma consideração. Somente nos últimos anos a idéia de integrar esse tipo de análise foi apoiada por novas metodologias da ER destinadas a preencher a lacuna entre o processo de desenvolvimento de *software* e a sua validação não-funcional (SHAH E PATEL, 2016).

A literatura apresenta um variado conjunto de categorias de RNF com poucas discrepâncias entre elas e que em sua maioria são congruentes entre si (AFZAL, TORKAR E FELD, 2008), A norma ABNT ISO/IEC 9126-1 (BRASIL, 2003), (DENNIS,

WIXOM E ROTH, 2014), (EL-HADARY E EL-KASSAS, 2014), (SOMMERVILLE, 2011), (PRESSMAN, 2001) apud (OLSINA, 1999), (BARBOSA et al, 2009), (SHAH E PATEL 2016). O que se aborda nessa Seção são exatamente essas variações e busca-se tais diferenças para contemplar o conjunto de RNF a serem utilizados, em forma de lista de *checklist*, para colimar com os objetivos desse estudo.

Diante do apresentado até o momento neste trabalho pressupõe-se que há uma série de RNF que são abordados na literatura que trata do assunto sem, no entanto, se chegar a um consenso padronizado em relação a eles. O Quadro 1 reúne todas os RNF identificados neste estudo com suas respectivas descrições e índices. Apresenta também suas descrições e métricas baseadas nos índices quando oportunas.

Acrescenta-se ainda que a aceitação, pela equipe desenvolvedora, de quais RNF devem ser consideradas, depende do tipo de *software* a ser construído, bem como da aprovação pelos *Stakeholders* envolvidos no processo de desenvolvimento. Portanto essa é uma atividade que merece ser tratada em conjunto, preferencialmente em reuniões técnicas do projeto com seus *Stakeholders*.

Para se alcançar a meta desejada quanto à qualidade do sistema, é imperioso não só atentar para quais RNF que serão considerados, mas principalmente se apresentam índices aceitáveis para cada atributo considerado, quando o sistema estiver em desenvolvimento e finalmente quando já for configurando no ambiente de produção.

Quadro 1: Requisitos Não Funcionais e suas métricas

RNF	Descrição	Métrica
Confiabilidade	Capacidade do <i>software</i> de manter seu nível de desempenho sob condições estabelecidas durante um período de tempo estabelecido (BRASIL, 2003).	Tempo de retorno de falhas (IZUKURA, 2015), em unidade de tempo
Continuidade	Tempo de inatividade (IZUKURA, 2015).	Unidade de tempo
Desempenho	Eficiência no tempo (IZUKURA, 2015).	Tempo de resposta em segundos
<i>Designer</i>	Projeto de desenvolvimento de sistema	a) Tempo de execução do projeto b) Custos disponibilizados
Disponibilidade	Disponibilidade do sistema em tempo útil,	Unidade de tempo disponível pelo

	restrições sobre janelas de manutenção, janelas de produção, soluções de contorno quando houver queda de energia etc.	total de tempo de um ano
Eficiência	Conjunto de atributos que evidenciam o relacionamento entre o nível de desempenho do <i>software</i> e a quantidade de recursos usados, sob condições estabelecidas (BRASIL, 2003).	
Escalabilidade	Taxa de transferência (IZUKURA, 2015).	MBytes/seg
Flexibilidade	Um programador de manutenção com pelo menos 6 meses de experiência no suporte ao produto deverá ser capaz de dar suporte a um outro dispositivo de interconexão em não mais do que 1 hora de trabalho	
<i>Hardware</i> ou Físico	Define as características de infraestrutura necessária ao bom funcionamento do SI no ambiente de produção.	<ul style="list-style-type: none"> a) Unidades de processamento incluindo-se coprocessadores em GHz b) Tamanho da memória principal em Gb ou Tb c) Tipos e tamanhos dos periféricos de armazenamento em Gb ou Tb d) Placas de expansão em Gb ou Tb e) Dispositivo de entrada e saída
Interface	Refere-se às características requeridas à interface do sistema.	
Implementação	Refere-se a restrições de implementação do <i>software</i> no ambiente de produção. Isso pode ser identificado como uma Regra de Negócio ao projeto.	
Interoperabilidade	Capacidade de o <i>software</i> poder ser operado em ambientes diferentes	
Manutenibilidade	Conjunto de atributos que evidenciam o esforço necessário para fazer modificações especificadas no <i>software</i> (BRASIL, 2003).	Custo monetário de manutenções
Portabilidade	Conjunto de atributos que evidenciam a	

	capacidade do <i>software</i> ser transferido de um ambiente para outro (BRASIL, 2003).	
Proteção	Qualidade de o <i>software</i> e hardware estão sob um ambiente seguro, ou seja, acessado por um público restrito	
Performance	Eficiência no tempo	Unidade de tempo
Reusabilidade	Facilidade de reuso de seus componentes, ou inserção de componentes externo ao sistema	
Robustez	Todas as variáveis de entrada terão valores <i>default</i> e tais valores serão usados sempre que dados de entrada estiverem faltando ou inválidos	Utilização de memória em Mb
<i>Software</i>	Define os requisitos mínimos para o funcionamento adequado do <i>software</i> . Por exemplo, se o sistema faz integração com o Microsoft Outlook, este deve estar instalado no computador como pré-requisito. Esse item também abrange a portabilidade do <i>software</i> para outros sistemas, tal como a sua facilidade de configuração.	Grau de entendimento do código fonte
Segurança	deve garantir a segurança dos dados, bem como as permissões de acesso às suas funcionalidades, como por exemplo, usar criptografia em senhas e liberar acesso aos menus do sistema de acordo com a hierarquia do usuário. Quando se trata de um <i>software</i> com informações confidenciais (como dados de vendas, faturamentos ou citações de pessoas), este item se torna indispensável. Refere-se a: a) Confidencialidade; b) Integridade. (IZUKURA, 2015).	a)Risco de vazamento de dados b)Risco de adulteração de dados c)Taxa de disponibilidade
Testabilidade	Capacidade de ter relativa facilidade em testar o sistema, ou mais importante ter	

	sido submetido exaustivamente a testes	
Usabilidade	Conjunto de atributos que evidenciam o esforço necessário para se poder utilizar o <i>software</i> , bem como o julgamento individual desse uso, por um conjunto explícito ou implícito de usuários (BRASIL, 2003).	

Fonte: os autores

5 Estudo de Caso

5.1 Objetivo do Estudo de Caso

Essa Seção do trabalho tratará de um estudo de caso para que se possa constatar se realmente a inclusão durante a fase de desenvolvimento, em particular na elicitação dos RNF, o mais abrangente possível, se realmente agrega mais qualidade do produto de *software* a ser desenvolvido. O estudo de caso a ser tratado tem como enfoque fazer um comparativo quanto ao emprego de RNF poderá afetar na qualidade de um mesmo sistema de TI, em versões diferentes.

5.2 Sistema Escolhido

O sistema escolhido é o Sistema de Cadastramento de Produtos de Defesa (SISCAPED), em produção no Ministério da Defesa (MD) do Brasil, atentando para o que se observou no sistema antigo, versão 1, e no novo sistema, versão 2, já em produção desde 2016. Assim poderá se inferir as mudanças e se realmente a aplicação de RNF com atenção ao maior número desses, se realmente houve melhoras quanto a qualidade do referido sistema.

O SISCAPED visa atender a uma demanda relativamente recente do MD quanto ao cadastramento de empresas e produtos considerados como estratégicos de defesa para o Brasil e apresentou uma solução automatizada deste processo, por meio de Tecnologia da Informação, consolidada em um sistema de informação.

Seu desenvolvimento ocorreu em duas versões distintas. A primeira não foram adotadas todas as práticas preconizadas na Engenharia de *Software* (ES) e Gerência

de Projeto (GP), já a segunda versão buscou-se adotar as boas práticas da ES e GP gerando um compêndio de documentação formulado por uma equipe multidisciplinar, visando implementar recursos funcionais inexistente e melhorar seu funcionamento de um modo geral. Tudo isso motivado pela equipe de desenvolvimento e seus usuários, as empresas que comercializam produto de interesse da defesa do Brasil.

5.3 Justificativa pela escolha

Por ser um sistema de grande importância para o MD e empresas que produzem ou fornecem produtos de interesse da defesa do Brasil, é crucial que funcione com boa qualidade fruto da quantidade de empresas que vem se cadastrando e que participam de licitações na venda desses produto para o MD, e também servem como uma base catalográfica, que em caso de guerra poderá o MD acionar tais empresas.

5.4 Descrição e Análise

Os procedimentos para a execução das avaliações dos dois sistemas foram considerados que ambos foram instalados no mesmo ambiente de produção, conforme apresentado na Figura 2, para tratar com a maior fidelidade a comparação desses dois sistemas.

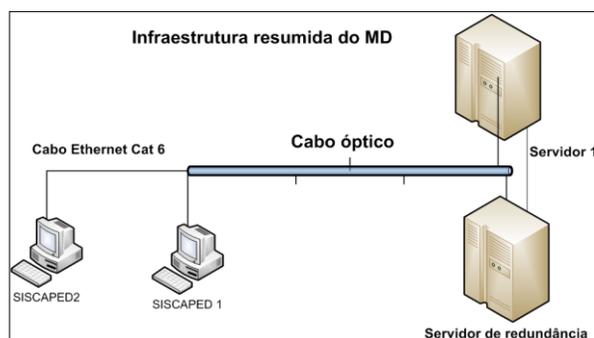


Figura 2: Infraestrutura utilizada para o Estudo de Caso.

Fonte: os autores

Para constatação da aplicação de RNF nos dois sistemas utilizou-se do *checklist* apresentado nesse estudo onde foram inseridos os quesitos RNF apresentados no Quadro 1 da sessão anterior e consolidadas suas observações no Quadro 2 abaixo apresentado. Os valores de índices subjetivos apresentados no Quadro 2 são valores de 0, pior resultado a 5, melhor resultado.

O preenchimento do Quadro 2 se deu por um Analista de Sistemas do MD e a medida em que os quesitos lhe era apresentado o mesmo pontuava de acordo com o limite de valores já definidos. Há de se considerar a subjetividade das questões presentes no *checklist*.

Quadro 2: Estudo comparativo envolvendo o SISCAPED versões 1 e 2

Nr	RNF	Índice a ser considerado	SISCAPED Versão 1	SISCAPED Versão 2
1	Confiabilidade	0 a 5	3	4
2	Continuidade	por unidade de tempo, hs, dias,etc	15 dias	15 dias
3	<i>Designer</i>	Tempo de execução do projeto	3 meses	3 meses
4	Disponibilidade		3	4
5	Eficiência	0 a 5	4	5
6	Flexibilidade	0 a 5	3	4
7	<i>Hardware</i> ou Físico	0 a 5	5	5
8	Interface	0 a 5	2	4
9	Implementação	0 a 5	1	1
10	Interoperabilidade	0 a 5	5	5
11	Manutenibilidade	0 a 5	2	2
12	Portabilidade	0 a 5	2	2
13	Proteção	0 a 5	4	4
14	Reusabilidade	0 a 5	1	1
15	Robustez	0 a 5	3	4
16		0 a 5	2	2

	Software			
17	Segurança	Todos de 0 a 5	3	4
		a) Risco de vazamento de dados	4	4
		b) Risco de adulteração de dados		
18	Testabilidade	0 a 5	2	2
19	Usabilidade	0 a 5	2	4
20	O sistema requer salvamento (backup) e recuperação (recovery) ?	0 a 5	2	4
21	Comunicações de dados são necessárias?	0 a 5	5	5
22	Há funções de processamento distribuídas?	0 a 5	0	0
23	O desempenho é crítico?	0 a 5	4	4
24	O sistema vai ser executado em um ambiente operacional existente, intensivamente utilizado?	0 a 5	5	5
25	O sistema requer entrada de dados on-line?	0 a 5	5	5
26	A entrada de dados on-line, exige que a transação de entrada seja construída através de várias telas ou operações?	0 a 5	5	5
27	Os arquivos mestres são atualizados on-line?	0 a 5	5	5
28	As entradas, saídas, arquivos ou consultas são complexas?	0 a 5	5	3
29	O processamento interno é complexo?	0 a 5	2	2
30	O código é projetado para se reusado?	0 a 5	2	2
31	A conversão e a instalação estão incluídas?	0 a 5	0	0
32	O sistema está projetado para instalações múltiplas em diferentes organizações?	0 a 5	4	4
33	A aplicação está projetada para facilitar modificações e para facilidade de uso pelo usuário?	0 a 5	3	4

Os quesitos de 20 a 33 do Quadro 2 apesar de estarem de certa forma embutidos em algum dos requisitos não funcionais eles expressão ainda a qualidade do *software* desenvolvido ou a ser desenvolvido. Por isso se torna importante para esse estudo.

Como se pode observar os resultados dos RNF que abrangeu o SISCAPED 2 foi consideravelmente maior em relação aos SISCAPED1, provendo maior qualidade do sistema e assim atendendo em melhores condições todos os usuários do sistema, sejam as empresas interessadas como os próprios usuários internos do MD.

A atribuição 2 no quesito manutenção para ambos sistemas é preocupante pois sistemas sempre estão sendo mantidos e o que se apresenta nesse quesito é que o sistema SISCAPED é de difícil manutenção.

Apesar de alguns índices de RNF para ambos sistema tenham apresentados resultados similares o maior número de aplicação de RNF no sistema 2, supõe-se indicar maior qualidade do *software* testado.

6 Conclusões e trabalhos futuros

O proposto nesse trabalho foi a construção de um *checklist* para, possivelmente, ser utilizado por futuros desenvolvedores de sistemas de TI e assim tentarem minimizar riscos nas fases iniciais de seu desenvolvimento, além de obter mais qualidade nesses sistemas de TI.

A criação de um *checklist* o mais completa possível serviu para aplicação no estudo de caso onde se constatou que no caso do SISCAPED 2 teve maior número de RNF atendidos, indicando uma melhora desse sistema em comparação com o SISCAPED 1. Isso nos leva a crer que o *checklist* produto desse trabalho servirá para empregos no desenvolvimento de futuros sistemas de TI

Como trabalhos futuros esse *checklist* poderá ser aplicado em outros sistemas de TI a serem desenvolvido e poder se constatar se realmente houve maior qualidade deles quando estiverem em produção. Essa constatação poderá ser percebida se

houver reduzida manutenção do sistema nas questões que envolvem infraestrutura utilizada por eles.

Information Systems Quality based on Non-Functional Requirements

Abstract

Many Information Systems have quality deficiencies simply because their developers do not attempt *with criteria based on Non-Functional Requirements needed to be achieved. The quality deficiency of software products is one factor that leads to fail many Information Technology Systems. The objective of this paper is to present a list, as complete as possible, of Non-Functional Requirements categories and its indexes, in order to checked quality in IT systems in futures IT projects.*

Keywords: *Quality; Non-Functional Requirement; Software Engineering.*

Referências

BARBOSA, Glívia; WERNECK, Marcelo; ASSIS, Helen; Ulisses & SILVA, Ismael (2009). "Um processo de elicitação de requisitos com foco na seleção da técnica de elicitação". Sociedade Brasileira de Qualidade de Software, VIII Simpósio Brasileiro de Qualidade de Software, Ouro Preto-MG, 2009.

BEZERRA, Eduardo. "Princípios de Análise e Projeto de Sistemas com UML.2ª Ed, Editora Campus. Rio e Janeiro-RJ, Elsevier, 2007, 369p.

BRASIL. Associação Brasileira de Normas Técnicas (ABNT) NBR ISO/IEC 9126-1 (2003). "Engenharia de Software - Qualidade de Produto, Parte 1: Modelo de Qualidade". ABNT, Rio de Janeiro-RJ, 2003.

ALSHAZLY, Amira A.; ELFATATRY Ahmed M., ABOUGABAL, Mohamed S. "Detecting defects in software requirements specification", Alexandria University, Alexandria Engineering Journal. Vol 53, p 513-527, 2014.

_____. Associação Brasileira de Normas Técnicas (ABNT) NBR ISO/IEC 14598-1 (2001). "Tecnologia de informação - Avaliação de qualidade de software". ABNT, Rio de Janeiro-RJ, 2001.

_____. Associação Brasileira de Normas Técnicas (ABNT) NBR ISO/IEC 12119 (1998) "Tecnologia de informação - Pacotes de software - Teste e requisitos de qualidade". ABNT, Rio de Janeiro-RJ, 1998.

DENNIS, Alan; WIXOM, Barbara Haley; ROTH, Roberta M. Análise e Projeto de Sistemas. Editora LTC, 5ª ed. Rio de Janeiro. 2014.

GRADY, Robert (1992). "Practical Software Metrics for Project Management and Process Improvement", Prentice-Hall, 1992.

Grupo Gartner. URL: em www.gartner.com. Acesso em nov 2016.

MONTEIRO, Mário A (1996). "Introdução à Organização de Computadores". 3ª Ed. Rio de Janeiro, LTC, 1996.

OLSINA, L., et al. "Specifying Quality Characteristics and Attributes for Web Sites". Proc. First ICSE Workshop on Web Engineering. ACM, Los Angeles, Mai 1999.

PFLIEGER, Shari Lawrence (2004). "Engenharia de Software - Teoria e Prática". Pearson Prentice Hall, 2ª ed., São Paulo, 2004.

PRESSMAN, Roger S. "Software Engineering – A Practitioner's Approach". McGraw-Hill, Fifth Edition, 2001.

ROCHA, José Gladistone da & RODRIGUES, Carlo Kleber da Silva. "Uma Proposta de Customização de Metodologia para o Desenvolvimento de Sistemas de Informação de Nível Acadêmico" Universitas em Gestão de TI, 2015. URL: <http://www.publicacoesacademicas.uniceub.br/index.php/gti>". Acesso em Jan 2016.

RUP, Rational Unified Process. URL: <http://www.wthreex.com/rup/portugues/index.htm>. Acesso em Jan 2016.

SHAHA Tejas, PATEL S. V. "Uma nova abordagem para a especificação de requisitos funcionais e não funcionais usando RDS (Esquema de Descrição de Requisitos)". Procedia Computer Science 79 (2016) 852 – 860 – 7th International Conference on Communication, Computing and Virtualization, 2016

SIADAT, Seyed Hossein; Song Minseok. Understanding Requirement Engineering for Context-Aware Service-Based Applications. School of Technology Management, Ulsan National Institute of Science and Technology, Ulsan, South Korea. Journal of Software Engineering and Applications , 2012, 5, 536-544.

SIELO. URL:http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-65552003000200007. Acesso em abr 2017.

SILBERSCHATZ, Abraham; GALVIN, Baer; GAGNE, Greg (2015). "Fundamentos de Sistemas Operacionais", 9ª edição. LTC, VitalBook file, 2015.

SOMMERVILLE, Ian (2011). "*Software Engineering*", Pearson, 9ª. ed., São Paulo, 2011.

STAIR, Ralph M.; REYNOLDS, George W. (2011). "Princípios de Sistemas e Informação ", 9ª edição. Cengage Learning. São Paulo, 2011. 590p.

STALLINGS, William (2010). "Arquitetura e organização de computadores, Pearson Pratices Hall, 8ª ed., 2010, 625p.

WIKIPÉDIA. URL: <https://pt.wikipedia.org>. Acesso em abr 2017