

Métricas para Sistemas de Tempo Real

Conceitos, Aplicações e Desafios.

Darlan Florêncio de Arruda
Escola Politécnica de Pernambuco
Universidade de Pernambuco - UPE
Recife, Brasil
dfa@ecomp.poli.br

Leonilson de Araújo Barboza
C.E.S.A.R
Centro de estudos e sistemas avançados do Recife
Recife, Brasil
Lab5@cin.ufpe.br

Resumo - Métricas de software tem o papel muito importante dentro da engenharia de software, especialmente na gerência de projetos de software. Percebe-se que as atividades de medição são aplicadas nos mais diversos tipos de projetos de TI, inclusive em projetos de sistemas de tempo real. Esse trabalho teve como objetivo realizar um levantamento bibliográfico com o intuito de identificar conceitos, aplicações e possíveis desafios relacionados ao uso de métricas de software, em especial projetos de sistemas de tempo real.

Palavras-Chave - Métricas de Software; Sistemas de Tempo Real; Desafios; Aplicações de Métricas.

I. INTRODUÇÃO

Métricas de *softwares* possibilitam realizar uma das atividades mais fundamentais do processo de gerenciamento de projetos - o planejamento [11] - e servem como suporte a medição em diversos tipos de atividades e aplicações. No processo de utilização de métricas de *software*, deve-se garantir algumas propriedades para que as mesmas sejam utilizada da maneira mais eficiente possível [11].

Guarizzo, (2008) [5] comenta que a utilização de métricas de *software* exerce um papel muito importante dentro da engenharia de *software*, especialmente na gerência de projetos de *software*, seja qual for a metodologia ou tipo a ser utilizada. Ela é analisada por gerentes de projetos de *software* e coletada pelos engenheiros de *software*.

Goodman, (1993) [4] define métricas de *software* como sendo, a contínua aplicação de técnicas baseadas na medição para o processo de desenvolvimento de *software* e para os seus produtos fornecendo informação de gestão relevante e, juntamente com a utilização dessas técnicas visando melhorar o processo e os seus produtos. Não seria diferente para as aplicações de tempo real.

De acordo com Farines et al., 2000 [2] na medida em que o uso de sistemas computacionais prolifera na sociedade atual, aplicações com requisitos de tempo real tornam-se cada vez mais comuns. Essas aplicações variam muito em relação à complexidade e às necessidades de garantia no atendimento de restrições temporais. Essas aplicações que apresentam a característica adicional de estarem sujeitas a restrições temporais, são agrupados no que é normalmente identificado como Sistemas de Tempo Real [2].

Esse trabalho teve como objetivo realizar uma revisão bibliográfica com o intuito de levantar os mais diversos tipos de métricas de *software* específicas e/ou úteis para sistemas de tempo real, além de identificar conceitos, aplicações e desafios relacionados ao uso de métricas de *software*. Sendo assim, esse trabalho apresenta os resultados iniciais e parciais identificados e colhidos com a revisão bibliográfica realizada.

O trabalho está organizado da seguinte forma: A próxima seção aborda sobre os procedimentos metodológicos aplicados na pesquisa. A seção 3 apresenta de forma geral os principais conceitos sobre métricas de *software*. A seção 4 discorre sobre sistemas de tempo real. Na seção 5 será possível identificar os tipos de métricas de *software* encontrados na literatura específica para sistemas de tempo real. A seção 6 discorre sobre as limitações e desafios que as métricas possuem. Por fim a seção 7 traz as conclusões e trabalhos futuros.

II. METODOLOGIA DE PESQUISA

Para Silva, (2001) [13] a metodologia científica é o conjunto de processos e operações mentais que se deve empregar nas investigações. É a linha de raciocínio adotado no processo da pesquisa. Ou seja, para que uma pesquisa seja efetuada é necessário um conjunto de procedimentos intelectuais e técnicos. Uma pesquisa pode ser classificada por diversas características, que podem ser, por exemplo, quantos aos fins e quanto aos meios.

Quanto aos fins, a presente pesquisa pode ser considerada descritiva, pois de acordo com Gil, (2009) [3] este tipo de pesquisa adota como objetivo primordial as descrições das características de uma determinada população ou determinado fenômeno ou o estabelecimento de relações entre variáveis.

Quanto aos meios, esse trabalho possui características de pesquisa bibliográfica. Segundo Silva, (2001) [13], pesquisa bibliográfica é elaborada a partir de material já publicado, constituído principalmente de livros, artigos de periódicos e atualmente com material disponibilizado na internet.

III. MÉTRICAS DE SOFTWARE

Métricas de *software* servem como suporte a medição em diversos tipos de atividades e aplicações, como por exemplo, utilização de métricas no contexto organizacional de gestão do

conhecimento, no apoio a sistemas baseados em computação em nuvem, no suporte a medição de complexidade do *software*, medição de esforço de trabalho, métricas no contexto de custos em manutenção corretiva de *software*, como suporte a mensuração de qualidade em aplicações de negócios, inspeção de *software*, métricas no contexto de qualidade e testes de *Software* e também métricas de *software* para sistemas de tempo real [2].

Sendo assim, percebe-se o quão abrangente é a atividade de métricas, e não somente em projetos de *software* mas em qualquer atividade a qual deseja-se medir, estimar e gerar métricas para uso no futuro. Para Yu e Zhou, (2012) [17] métricas de *software* é a medida - geralmente usando classificações numéricas - para quantificar algumas características ou atributos de uma entidade de *software*. Medições típicas incluem a qualidade dos códigos de fonte, o processo de desenvolvimento e as aplicações realizadas.

Já para Singh et al., (2012) [16], métricas de *software*, medem diferentes aspectos da complexidade do *software* e, portanto, desempenham um papel importante na análise e melhoria da qualidade do *software*. Tais aspectos, abrangem área de qualidade, estimativa, custos, processos e assim por diante. Sua aplicação é bem diversificada, podendo também ser aplicadas em projetos de sistemas de tempo real

De acordo com SINGH et al. (2011) [15], as métricas de *software* são definidas em dois tipos, a métricas de produto e as métricas de processos.

Métricas de Produto, também são conhecidos como as métricas de qualidade e são usadas para medir as propriedades do *software* e ajudam a melhorar a qualidade dos diferentes componentes e existentes do sistema [15]. Pode-se destacar como exemplos de métricas de produto: Métrica de esforço, métrica de tamanho, métrica de volume, métrica de complexidade, contagem de pontos de função, contagem de *Tokens*, métricas de funcionalidades, métricas de desempenho, métricas de usabilidade, métricas de custo e tamanho e métricas de estilo, por exemplo.

Já as métricas de processo são conhecidas como métricas de gestão e utilizados para medir as propriedades do processo que é usado para se obter o *software*. Métricas de processo incluem as métricas de custo e esforços, métricas de progresso e métricas de reutilização. Métrica de processo ajuda na previsão do tamanho do sistema final e determinar se um projeto sobre a execução de acordo com o cronograma [15]. Como métricas de processos podemos citar: métricas de reusabilidade, métricas de produtividade, Métrica Média de ponto de estória por desenvolvedor por dia, métricas de recursos, dentre outros.

Comparar e avaliar as capacidades e produtividade das pessoas envolvidas no desenvolvimento de *software*, elaboração das especificações de qualidade de *software*, verificação do cumprimento de requisitos de sistemas de *software* e especificações, qualidade do produto desenvolvido, tomada de decisões sobre outra divisão do módulo complexo que deve ser feito ou não e obtenção de um ideia sobre a complexidade do *software* - são algumas das características de

métricas quando utilizadas de forma correta [15,16]. Essas características também são considerados pelos autores citados [15,16] como vantagens que a aplicação de métrica possibilita a quem a usa.

Ao mesmo tempo em que as métricas de *software* possuem certas vantagens que ajudam no auxílio de algumas atividades dentro do processo de desenvolvimento, também possuem certas limitações em relação ao seu uso. Pode-se destacar como sendo limitação, o fato da aplicação de métricas de *software* nem sempre é fácil e, em alguns casos, pode ser difícil e dispendioso. A verificação e justificação de métricas de *software* é baseada em dados históricos / empírica, cuja validade é difícil de verificar. A maioria dos modelos de previsão baseiam-se em estimativas de certas variáveis que muitas vezes não são conhecidas exatamente. São úteis para o gerenciamento de produtos de *software*, mas não para avaliar o desempenho da equipe técnica [15].

As métricas devem ser facilmente entendidas, calculadas e testadas, devem gerar uma sugestão de melhoria nas estratégias, bem como é de grande importância que a métrica seja obtida o mais cedo possível no processo de desenvolvimento do *software* [1].

A seção a seguir discorre sobre sistemas de tempo real.

IV. SISTEMAS DE TEMPO REAL

Para Farines et al., 2000 [2], na medida em que o uso de sistemas computacionais prolifera na sociedade atual, aplicações com requisitos de tempo real tornam-se cada vez mais comuns. Esses sistemas variam muito em relação à complexidade e às necessidades de garantia no atendimento de restrições temporais [2]. Essas aplicações que apresentam a característica adicional de estarem sujeitas a restrições temporais, são agrupados no que é normalmente identificado como Sistemas de Tempo Real [2].

Atualmente, na literatura existem diversas definições para sistemas de tempo real. Percebe-se que muitos pesquisadores deram sua contribuição e tentaram definir conceitos que melhor representassem essa tecnologia. A seguir serão apresentadas algumas dessas definições.

Um Sistema de Tempo Real é definido por Prado (2010) [12] como o *software* que gerencia os recursos de um sistema computacional, com o objetivo de garantir com que todos os eventos sejam atendidos dentro de suas restrições de tempo, e gerenciados da forma mais eficiente possível.

Para Farines et al (2000) [2], Um Sistema de Tempo Real (STR) pode ser definido como um sistema computacional que deve reagir a estímulos oriundos do seu ambiente em prazos específicos. Toda tarefa computacional recebe um evento, que pode ser interno ou externo, realiza o processamento e produz uma saída e essa saída é gerada dentro de um limite de tempo máximo estabelecido [12].

De acordo com Shaw [2001] [19] Sistemas de tempo real são sistemas de computação que monitoram, respondem e controlam um ambiente externo [19]. Já Mall (2009) [10] primeiramente define o conceito de tempo real para em seguida

definir e entender o termo sistemas de tempo real. Dessa forma Mall [2009] coloca que tempo real é a noção quantitativa de tempo e é mensurada usando um relógio físico (real). Um sistema é dito de tempo real quando precisamos expressar de forma quantitativa de tempo para descrever o comportamento do sistema [10].

Os sistemas de tempo real podem ser classificados a partir do ponto de vista da segurança, em sistemas não críticos de tempo real, quando as consequências advindas de uma falha devida ao tempo é dada na ordem de grandeza que os benefícios do sistema em operação normal e os sistemas críticos de tempo real, quando as consequências de pelo menos 1 falha temporal excedem em muitos benefícios normais do sistema, nesse caso a falha é dita catastrófica [2].

Aplicações exemplos de sistemas de tempo real são onipresentes e estão proliferando, aparecendo como parte de nossas infraestruturas comerciais, governamentais, militares, médicas, educacionais e culturais. São exemplos de sistemas de tempo real [19]:

- Sistemas de controle de veículos para automóveis, metrô, aeronaves, ferrovias e navios;
- Controle de tráfego para autoestradas, espaço aéreo, trilhos de ferrovias;
- Controle de processos para usinas de energia, indústrias químicas
- Sistemas médicos para radioterapia, monitoramento de pacientes.

Restrições temporais, novos critérios de regularidade, sistema de computação embutido, segurança crítica, concorrência, estrutura distribuída, criticidade de tarefas, hardware customizado, reatividade e estabilidade são alguns dos atributos que caracterizam e diferem sistemas de tempo real dos demais tipos de *software* [10].

A seção a seguir discorre acerca de aspectos referentes a métricas de *software* para sistemas de tempo real, bem como apresenta quadros com alguns tipos de métricas para sistemas de tempo real identificados na literatura.

V. MÉTRICAS DE SOFTWARE PARA SISTEMAS DE TEMPO REAL

Uma vasta gama de métricas de *software* visando vários níveis de abstração e atributos de qualidade têm sido propostas pela comunidade de pesquisa [18]. Métricas podem ser usadas em sistemas de tempo real de diversas formas, que abrange desde a especificação dos requisitos dando suporte a atividades de estimativas de custos e passando por outra aplicação bastante útil que seria a de *benchmarking*¹ [9].

Este trabalho não fornece uma *guideline* sobre as aplicações de métricas, o objetivo central é o levantamento delas. Quando se fala em métricas para sistemas de tempo real, percebe-se que elas possuem uma importância grande quando se trata em

¹ *Benchmarking* é o processo contínuo de medição de produtos, serviços e práticas em relação aos mais fortes concorrentes, ou às empresas reconhecidas como líderes em suas indústrias (CAMP, 1998, p. 8 e 10)"

medir atributos de qualidade desses tipos de sistemas. Nessa seção buscou-se agrupar o maior número de métricas identificadas nas publicações pesquisadas com o intuito de contextualizar a acerca dos tipos de métricas disponíveis para medição e garantia de qualidade em sistemas de tempo real.

O quadro abaixo mostra de forma sucinta tais métricas identificadas. Esse quadro está organizado da seguinte forma: Três colunas contendo os campos nome da métrica, breve descrição e a referência aos autores dos trabalhos onde a métrica foi identificada.

QUADRO 1. MÉTRICAS PARA SISTEMAS DE TEMPO REAL

Métrica	Breve Descrição	Referência
Tempo de Chaveamento entre duas tarefas	Este tempo inclui salvar os registradores da tarefa que está executando e carregar o registradores com os valores da nova tarefa, incluindo qualquer informação necessária para a MMU (<i>"memory management unit"</i>) funcionar corretamente. Em geral esta métrica não inclui o tempo necessário para decidir qual tarefa vai executar, uma vez que isto depende do algoritmo de escalonamento utilizado.	[FARINES et al., 2000].
Tempo de Latência	Essa métrica mede a latência até o início do tratador de uma interrupção do <i>hardware</i> . A latência no disparo de um tratador de interrupção inclui o tempo que o <i>hardware</i> leva para realizar o processamento de uma interrupção, isto é, salvar o contexto atual (<i>"program counter"</i> e <i>"flags"</i>) e desviar a execução para o tratador da interrupção.	
Latência de Interrupções	Cálculo que corresponde ao intervalo de tempo que decorre entre o pedido de interrupção e o início ao seu atendimento	
Tempo de Resposta	Essa métrica tem por objetivo realizar a medição do tempo de respostas a eventos internos e externos da aplicação.	[JONSSON, 1997]
Tempo total de resposta	Cálculo do tempo entre a sinalização do evento através de uma interrupção de <i>hardware</i> e a respectiva resposta do sistema ao ambiente.	
Proporção de flacidez normalizada (<i>NORM</i>)	É o cálculo da razão entre a flacidez geral para a soma dos tempos de execução de todas as tarefas num caminho X. Com esta métrica, flacidez é atribuída em proporção ao tempo de execução da tarefa.	
Proporção de laxismo puro (<i>PURE</i>)	É a razão entre a flacidez geral com o número de tarefas, n, num caminho. Para essa medida, à todas as tarefas são atribuídas uma parcela igual de flacidez.	
Perfil de Performance (<i>Performance profiling</i>)	Este é um meio para determinar o tempo que um sistema gasta de função a função. Permite os desenvolvedores a identificar áreas problemáticas e focar nas seções da aplicação onde o benefício máximo seria obtido a partir dos esforços de otimização de desempenho.	(HILLARY, 2005)
Tempo A-B	Usada para mensurar o tempo gasto	

(A-B Timing)	para ir de um ponto específico do código a outro. Isso é talvez, a mensuração de tempo mais importante para os sistemas de tempo real, uma vez que permite aos desenvolvedores verificar se os objetivos de tempo de um pedaço do código estão sendo satisfeitos.	(HILLARY, 2005)	
Resposta a eventos externos	É um meio de medir o tempo gasto entre um evento externo ocorrer e o software responder a esse evento (Ex: Período de Latência de Interrupção). Esta medida é especialmente útil para os desenvolvedores do sistema em tempo real para quem os componentes de <i>hardware</i> são utilizados para garantir o desempenho do <i>software</i> .		
Linhas de Código	Essa métrica faz a contagem da quantidade de linhas de código disponíveis na aplicação, excluindo comentários e espaços em branco.		[Singh et al. 2012] [LAPLANTE, 2004]
Métricas de McCabe	Métrica usada para mensuração de fluxos de controle para medir complexidade ciclossomática do software.		[LAPLANTE, 2004]
Métricas de Halstead	Essa métrica mensura o conteúdo da informação e/ou a intensidade do uso da linguagem de programação		
Pontos de Recursos	Essa métrica é uma extensão da métrica pontos de função. A métrica de ponto de recurso é calculado de uma maneira semelhante ao ponto de função, exceto pela inclusão de um novo fator que possuem pesos empíricos e são distribuídos de 1 a 7 (w1...w7).		

	definidos em uma classe em um programa.	[LAPLANTE, 2004]
Número de Atributos	Métrica para contagem do número de atributos de uma classe.	[Singh et al. 2011]
Métrica de Acoplamento entre classes e objetos	É a contagem de todas as classes com que determinada classe está acoplada.	[Yu e Zhou, 2010]
Falta de Coesão de Métodos (LCOM)	Essa métrica usa variável ou atributos para medir o grau de semelhança entre os métodos.	[Singh et al. 2011]
Profundidade de árvore de herança (DIT)	Profundidade de classe dentro da hierarquia herança é o comprimento máximo do nó de classe para a raiz da árvore, medida pelo número de classes antepassado.	
Número de Filhos (NOC)	O número de filhos é o número de subclasses subordinados imediatos de classe na hierarquia.	

Foi possível através da literatura identificar diversos tipos de métricas que são úteis para projetos de sistemas de tempo real e que se usadas de forma correta, irão contribuir para a qualidade do produto final desenvolvido, mas não de acordo com os tipos específicos de arquitetura e sim em uma abrangência geral. A seção a seguir apresentar as limitações e desafios que são encontrados com a aplicação de métricas de *software*.

VI. DESAFIOS ENCONTRADOS NA APLICAÇÃO DE MÉTRICAS DE SOFTWARE EM SISTEMAS DE TEMPO REAL

Quando se pensa em produtos de *software*, o grande desafio é estimar custos, esforço, tamanho e prazo para criação de métricas que realmente representem a realidade. Realizar estimativas e prover métricas tem se tornado uma incapacidade da indústria de TI [11]. Sendo assim, essa seção tem como objetivo elencar os principais desafios hoje enfrentados quando se pensa em aplicar métricas em projetos de *software* e em especial projeto de sistemas de tempo real.

Através das revisões bibliográficas realizadas não foi possível identificar desafios específicos com aplicação de métricas de *software* para sistemas de tempo real, entretanto, muitas limitações e desafios, foram identificados e através dos dados observados, pode-se inferir que tais desafios, também devem ser considerados quando se pensa em utilizar métricas de *software* para sistemas de tempo real.

Alguns fatores considerados desafios relacionados a aplicações de métricas de *software*, podem ser identificados em diversas publicações no meio acadêmico. A seguir são apresentados alguns desses fatores que são considerados limitações e desafios para a referida área [11, 15, 16].

- A arquitetura do sistema pode ser um fator complicante para o uso de métricas em sistemas de tempo real.

Além das métricas apresentadas no quadro 1, foram identificadas outros tipos de métricas, em específico, as métricas de *software* orientados a objetos que segundo Lenplate [9], esses tipos de métricas ajudam na melhoria da consistência da aplicação de tempo real quando desenvolvida no paradigma de orientação a objetos. Não existem particularidades dessas métricas para sistemas em tempo real, mas devido ao paradigma OO ser bastante utilizado, inclusive em sistemas de tempo real, achou-se importante incluí-las no presente trabalho.

O quadro abaixo traz o mapeamento dessas métricas e está organizado da seguinte forma: Três colunas contendo os campos nome da métrica, breve descrição e a referência aos autores dos trabalhos onde a métrica foi identificada.

QUADRO 2. MÉTRICAS DE OO PARA SISTEMAS DE TEMPO REAL

Métrica	Breve descrição	Referência
Número de Classe	Métrica para contagem do número de classes em um programa.	[Johari e Kaur, 2011]
Número de pacotes	Métrica para contagem do número de pacotes em um programa.	[LAPLANTE, 2004]
Número de Métodos por classe	Métrica para contagem do número de métodos	[Johari e Kaur, 2011]

- A aplicação de métricas de *software* nem sempre é fácil e, em alguns casos, pode ser difícil e dispendioso.
- A verificação e justificação de métricas de *software* é baseada em dados históricos / empírica, cuja validade é difícil de verificar.
- São úteis para o gerenciamento de produtos de *software*, mas não para avaliar o desempenho da equipe técnica.
- A definição e derivação de métricas de *software* é geralmente baseada em assumir que não são padronizados e podem depender de ferramentas disponíveis e ambiente de trabalho.
- A maioria dos modelos de previsão basear em estimativas de certas variáveis que muitas vezes não são conhecidos exatamente.

Além desses fatores elencados acima, os autores desse trabalho acreditam que a incapacidade de produzir requisitos precisos, deve-se à falta de experiência em projetos similares, falta de experiência e de conhecimento do profissional além da constante evolução da TI são desafios muito importantes que devem ser tratados para que atividades de medição e estimativa sejam realizadas da melhor forma possível, seja para projetos de sistemas reais ou qualquer outro de tipo de projeto de *software*.

VII. CONCLUSÕES E TRABALHOS FUTUROS

Esse trabalho abordou métricas de *software* num contexto organizacional de projetos de sistemas de tempo real. Através da revisão bibliográfica foi possível identificar características importantes sobre a aplicação de métricas de *software* em sistemas com esse paradigma.

Foi possível perceber através do levantamento realizado, os mais diversos tipos de métricas disponíveis na literatura para aplicação em projetos de sistemas de tempo real. Foi criado quadros que mostraram de forma sucinta essas métricas e suas definições. Pôde-se identificar métricas para qualidade e também métricas de projetos de orientação a objetos que foram identificadas como importantes por alguns dos autores citados nesse trabalho pelo fato de ajudarem na melhoria da consistência da aplicação.

Também ficou evidente que as métricas de *software* proporcionam diversas vantagens para que as usam, porém apesar de possibilitar diversas vantagens existem muitos desafios que devem ser superados para que as atividades de medição sejam utilizadas em sua plenitude.

Como sugestões de trabalhos futuros, destaca-se a necessidade de ampliar o estudo através de análises mais profundas acerca dos tipos de métricas para sistemas de tempo real, elicitando seus desafios e possíveis propostas de solução,

além de analisar outros desafios que por hora não são os mais importantes para o desenvolvimento da área, mas, que possam contribuir para aprimorar os estudos na área e enriquecer o seu conteúdo teórico.

REFERÊNCIAS

- [1] COSTA et al 2013. Utilizando métricas para dimensionar um software. Disponível em: <http://www.csi.uneb.br/engenharia_de_software/anexos/Artigo>MetricasdeSoftware.pdf>. Acessado em 20 de Julho de 2013.
- [2] FARINES, Jean-Marie; FRAGA, Joni da Silva; OLIVEIRA, Rômulo Silva de. Sistemas de Tempo Real. Disponível em: <<http://www.das.ufsc.br/~romulo/livro-tr.pdf>>. Acesso em 02 de Agosto de 2013.
- [3] GIL, A. C. Métodos e técnicas de pesquisa social. 6. ed. São Paulo: Atlas, 2009
- [4] GOODMAN, Paul. Practical Implementation of Software Metrics, McGraw Hill, London, 1993.
- [5] GUARIZZO, Karina. Métricas de Software. Trabalho de conclusão de curso. Faculdade de Jaguariúna. 2008.
- [6] HILLARY, Nat. Measuring Performance for Real-Time Systems. Freescale semiconductor, INC, 2005.
- [7] JONSSON, Jan. A Robust Adaptive Metric for Deadline Assignment in Heterogeneous Distributed Real-Time Systems. 1997. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.53.260&rep=rep1&type=pdf>>. Acessado em 15 Julho de 2013.
- [8] JOHARI, Kalpana; KAUR, Arvinder. Effect of software evolution on software metrics: An open source case study. ACM SIGSOFT Software Engineering Notes, 2011.
- [9] LAPLANTE, Phillip A. Real-Time Systems Design and Analysis. Instituto de Engenharia Elétrica e Eletrônica, 3ª Ed, 2004.
- [10] MALL, Rajib. Real-Time Systems: Theory and Practice. Pearson, 2009.
- [11] MOLØKKEN, Kjetil; JØRGENSEN, Magne; A Review of Surveys on Software Effort Estimation. Proceedings of the 2003 International Symposium on Empirical Software Engineering, ISESE '03, IEEE Computer Society, pp.223-230, 2003.
- [12] PRADO, Sergio. Sistemas de tempo real - Parte 1. Disponível em: <<http://sergioprado.org/sistemas-de-tempo-real-part-1/>>. Acessado em 30 de Julho de 2013.
- [13] SILVA, Edna Lúcia. Metodologia da pesquisa e Elaboração de dissertação. 3ª Ed. Florianópolis: 2001.
- [14] SHAW, Alan C. Sistemas e Softwares de Tempo Real. Bookman, 2001.
- [15] SINGH, Gurdev; SINGH, Dilbag; SINGH, Vkram. A Study of Software Metrics. IJCEM International Journal of Computational Engineering & Management, Vol. 11, January 2011.
- [16] SINGH, Ram; BHAGAT, Avinash; KUMAR, Navdeep. Generalization of Software Metrics on Software as a Service (SaaS). International Conference on Computing Sciences, 2012.
- [17] YU, Sheng; ZHOU, Shijie. A Survey on Metric of Software Complexity. Information Management and Engineering (ICIME), 2010
- [18] BOUWERS, Eric; DEURSEN, Arie Van; VISSER, Joost. Evaluating Usefulness of Software Metrics - an Industrial Experience Report - .Software Engineering Research Group, Department of Software Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2013.
- [19] SHAW. Alan C. Real-time Systems and Software. New York: John Wiley & Sons, 1. ed. 2001.