

Métricas de Software: Por que é difícil usá-las de forma eficiente?

Darlan Florêncio de Arruda¹, José Gilson de A. T. Filho¹, Ricardo de Lima Chen²

dfa@ecomppoli.br, jgatif@upe.poli.br, ricardolimachen81@gmail.com

¹Escola Politécnica da Universidade de Pernambuco – POLI/UPE, Brasil
Rua Benfica, Madalena, s/n.
Recife - Brasil.

²Faculdade dos Guararapes – FG, Brasil.
R. Comendador José Didier, N° 27, Piedade
Recife – Brasil.

Resumo: *Métricas de Software é um assunto que vem sendo estudado há anos e mesmo assim, ainda hoje desperta interesses de pesquisadores. Talvez pelo fato de ainda não ter atingido sua maturidade. Sabe-se que realizar estimativas e prover métricas de software eficientes tem se tornado um grande desafio na área de TI e que essa incapacidade da indústria em estimar software com precisão, resulta em derrapagens orçamentais e atrasos nas entregas. Diante disso, esse trabalho tem como objetivo investigar o estado da arte sobre métricas de software com o intuito de identificar possíveis causas que contribuem para o uso ineficiente de métricas. Sendo assim foi aplicado o processo de revisão sistemática da literatura com o intuito de responder a algumas questões de pesquisa. Na fase inicial da revisão sistemática a String de busca aplicada retornou 559. Após aplicação dos critérios de seleção nas fases da condução da revisão, apenas 22 trabalhos foram selecionados.*

Palavras chave: Revisão Sistemática da Literatura, Métricas de Software, Limitações, Benefícios.

Abstract: *Software Metrics is a subject that has been studied for years and still arouses interests of researchers. Perhaps because it has not reached its maturity yet. It is known that make estimates and provide efficient software metrics has become a major challenge in the IT field and that industry's inability in accurately estimating software, resulting in budget overruns and delays in deliveries. Therefore, this study aims to investigate the state of the art on software metrics in order to identify possible causes that contribute to the inefficient use of metrics. Therefore, we applied the systematic literature review process in order to answer some research questions defined for this paper. In the initial stage of the systematic review, the string returned 559 papers. After applying the selection criteria in the phases of the conduct of the review, only 22 studies were selected.*

Keywords: Systematic Review of the Literature, Software Metrics, Limitations, Benefits.

1 Introdução

Métricas de *software* é um assunto estudado há mais de 30 anos. No entanto, mal conseguiu se estabelecer em sua amplitude na engenharia de *software* [MOLØKKEN, 2003]. A principal razão para isto, é que a maioria das atividades de métricas de *software* não têm apresentado o requisito mais importante: fornecer informações para apoiar a tomada de decisão gerencial quantitativa, durante o ciclo de vida do *software* [FENTON e NEIL, 2000].

De acordo com [Singh *et al* 2012], 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.

Para [Yu e Zhou, 2012] a 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 Goodman [1993] 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.

Sendo assim, o uso de métricas se mostra de importância e relevância para os gestores e analista de projetos de *software*. Entretanto realizar estimativas e prover métricas de *software* tem se tornado um grande desafio na área de TI. A incapacidade da indústria em estimar *software* com precisão, resulta em derrapagens orçamentais e atrasos nas entregas que minam em confiança e desgaste. Estimar tamanho de um *software* é uma atividade complexa, cujos resultados devem ser constantemente atualizado levando em consideração quantitativos reais de todo o ciclo de vida [MOLØKKEN, 2003].

A construção de uma solução que de fato produza resultados significativos e que seja de fácil aplicação e entendimento ainda está sendo buscada, inclusive por empresas de TI [MOLØKKEN, 2003]. Sendo assim, métricas de *software* se mostra como um assunto relevante e que merece certa atenção em relação a seus conceitos, características e uso.

Visando entender os conceitos gerais e motivos que talvez contribuam para esse déficit na qualidade de uso das métricas em projetos de *software*, tem-se como objetivo responder as questões de pesquisa:

(RQ1) *Qual o estado da arte atual em relação estudos sobre métricas de software?*

(RQ2) *Quais as vantagens com o uso de métricas em projetos de software? Existem limitações com essa abordagem?*

(RQ3) *Quais o tipos de métricas de software identificadas com revisão da literatura?*

(RQ4) *Existem ferramentas que dão suporte ao uso de métricas de software?*

(RQ5) *Por que é difícil para a indústria de software realizar estimativas e prover métricas de forma eficiente?*

Para alcançar o objetivo desse trabalho, foi aplicado o método de revisão sistemática da literatura com o intuito de conseguir as respostas para esse questões de pesquisa. A aplicação da *String* de busca nas bases científicas selecionadas resultou em 559 trabalhos, dentre os quais foram classificados após etapas de seleção, 22 trabalhos relevantes para a pesquisa.

O restante do trabalho está organizado da seguinte forma: Na seção 2 é abordado o processo da revisão sistemática da literatura. Na seção 3 são descritos e comentados os resultados alcançados com a pesquisa. E por fim, a seção 4 traz as conclusões e trabalhos futuros.

2 Revisão Sistemática da Literatura

A revisão de literatura é o processo central que apoia todo projeto de pesquisa, permitindo que o conhecimento científico seja identificado de forma a possibilitar uma pesquisa planejada, evitando esforços duplicados e repetição de erros anteriores [DYBA et al, 2005]. Assim, se a revisão de literatura não for conduzida de uma forma confiável e abrangente, os resultados possuirão pouco valor científico, uma vez que ela pode ter sido guiada por interesses pessoais, ocasionando resultados pouco confiáveis [MAFRA E TRAVASSOS, 2006].

Ainda Segundo [Mafra e Travassos, 2006], a revisão sistemática atua como um meio para identificar, avaliar e interpretar toda pesquisa relevante e disponível sobre uma questão de pesquisa específica, tópico ou fenômeno de interesse, fazendo uso de uma metodologia de revisão que seja confiável, rigorosa e que permita auditoria.

Estabelece um processo formal para conduzir a investigação, evitando a introdução de vieses da revisão de literatura informal, dando maior credibilidade à pesquisa em andamento [SAMPAIO E MANCINE, 2007].

Dessa forma, a revisão sistemática pode ser melhor entendida como uma sintetização de resultados obtidos do estado da arte da literatura, sendo um tipo de estudo retrospectivo e secundário, isto é, a revisão é usualmente desenhada e conduzida após a publicação de muitos estudos experimentais (primários) sobre o tema [SAMPAIO E MANCINE, 2007].

O caminho trilhado na revisão sistemática envolve uma série de atividades importantes, atreladas a um conjunto de fases dentro do processo de condução, estabelecidos dentro de um protocolo, que norteará de forma sistematizada todo o processo de condução da revisão. Três fases do processo de condução: planejamento da revisão, execução da revisão e análise e divulgação dos

resultados. Outros autores abordam as atividades de forma um pouco diferente, porém convergentes [BIOLCHINI et al, 2005; PAI et al., 2004; LITTEL et al., 2008].

As fases de condução da revisão e as atividades utilizadas neste trabalho foram adaptadas dos trabalhos citados anteriormente e estão detalhadas na figura a seguir, incluindo ainda um comparativo entre uma revisão sistemática e uma revisão de literatura comum.

Fases do processo de condução	Atividades	Revisão Sistemática	Revisão de Literatura Comum
Planejamento da revisão	Formulação da questão	Questão focada no objetivo da pesquisa (ideia-tema)	Questão focada no objetivo da pesquisa, embora permita uma questão mais abrangente, fora da ideia-tema
	Identificação de estudos	Uso de estratégia de pesquisa transparente, que leva a resultados embasados cientificamente	Sem uma estratégia definida, tornando os resultados menos relevantes
	Avaliação crítica dos estudos	Critérios específicos de inclusão e exclusão são utilizados para avaliação	De forma geral, os estudos identificados não são avaliados de acordo critérios
Execução da revisão	Extração dos dados	Utilização de formulários padronizados para catalogação e avaliação posterior desses dados	Geralmente os dados são extraídos de forma singular, sem a utilização de um modelo padrão
Análise e divulgação dos resultados	Sintetização dos resultados	Os dados são analisados de forma descritiva, levando-se em conta a qualidade dos estudos, amostras extraídas e outros fatores que possam intervir nos resultados	Os dados são avaliados de forma isolada, sem levar em conta a qualidade dos estudos nos quais foram extraídos os dados
	Interpretação dos resultados	Os resultados interpretados estão menos sujeitos a desvios do sentido original dos estudos e opiniões pessoais invasivas	A interpretação dos resultados tende ao acúmulo de desvios do sentido original dos estudos obtidos, além de acumular opiniões pessoais

Figura 1: Fases, atividades e comparativo do processo de revisão sistemática e revisão da literatura comum [Pai et al., 2004; Littel et al., 2008]

As subseções a seguir apresentarão as etapas e o processo de condução da revisão sistemática aplicada nesse trabalho.

2.1 Condução da Revisão Sistemática da Literatura

O processo de revisão sistemática foi iniciado a partir da criação de um modelo de protocolo de revisão, onde no mesmo constam informações essenciais para a realização da revisão sistemática. O protocolo foi especificado para identificar os estudos potenciais que possam contribuir para a realização do objetivo desse trabalho descrito no inicialmente.

Na fase de planejamento foram estabelecidos os objetivos da pesquisa e foi criado um protocolo de revisão adaptado do modelo de Biolchini et al. (2005) e Kitchenham et al. (2004), contendo itens como: identificação e seleção das bases de dados, métodos de busca e palavras-chave, estratégia de busca e critérios de inclusão e exclusão dos estudos (Biolchini et al., 2005; Kitchenham et al., 2004; Dyba et al., 2005), seguindo ainda a mesma lógica de

atividades propostas e adaptadas por Pai et al. (2004) e Littell et al. (2008).

Devido ao foco da pesquisa abranger um conteúdo de relevância mundial, evidentemente as palavras-chaves foram formuladas no idioma universal Inglês. Os termos e sinônimos para as palavras-chave identificadas são apresentadas a seguir.

Termo: Métricas em projetos de *software* (**Sinônimos:** *Metrics, Software Metrics, Software estimation Metrics, Software Indicators, Software Development Metrics, Software Project Metrics, Software Engineering Metrics*)

Termo: Empresas de TI (**Sinônimos:** *IT Companies, IT Consulting*)

Com base nos sinônimos identificados e definidos anteriormente, buscou-se estabelecer as *strings* de buscas. Essas *strings* serão aplicadas de acordo com a disponibilidade técnica de estratégia de busca do mecanismo a ser utilizado, podendo sofrer pequenas adaptações para que o mecanismo consiga executá-las. A *String* definida é apresentada a seguir.

((Metrics OR "Software Metrics" OR "Software estimation metrics" OR "Software Indicators" OR "Software Development Metrics" OR "Software Projects Metrics" OR "Software Engineering Metrics") AND ("IT Companies" OR "IT consulting"))

Essa *string* de busca definida passou por testes pilotos e foi aprimorada de acordo com os resultados obtidos, sendo esta, a quarta versão da *String* – aprimorada e restrita para o objetivo dessa pesquisa: Levantar estudos significativos realizados na indústria (empresas e consultorias de TI) com o objetivo de responder as questões de pesquisas definidas para esse trabalho.

A aplicação da *string* de busca foi realizada nas seguintes fontes de busca: *IEEE Xplorer, Sciece Direct, Scopus, El Compendex e ACM Digital Library*.

Após realização das buscas com a *string* definida, os trabalhos foram selecionados seguindo critérios de seleção específicos. São eles:

- Artigos Científicos publicados entre 2008 e 2013;
- As fontes devem disponibilizar os trabalhos na íntegra;
- Devem ser escritos em inglês;
- Para trabalhos que representam os mesmo resultados de pesquisa, será aceito o trabalho que apresentar os dados de forma mais completa.

Foram excluídos os trabalhos que:

- Não satisfizeram os critérios de inclusão;
- Teses e dissertações que não foram disponibilizadas em forma de artigo.

A aplicação da *string* de busca inicialmente resultou em 559 trabalhos retornados e que já satisfizeram os critérios de inclusão inicialmente estabelecidos. Em seguida iniciou-se a seleção dos trabalhos analisando inicialmente o título, onde os estudos que tiveram o seu título

considerado relevante ao contexto da pesquisa foram potencialmente selecionado para a próxima etapa, e os demais excluídos. Essa etapa resultou em 88 artigos selecionados.

Em seguida, os estudos pré-selecionados na etapa anterior tiveram seus resumos (*abstract*) e palavras chaves lidos, depois foram selecionados para a próxima etapa e fichados no formulário de condução da revisão os estudos considerados relevantes, os demais foram excluídos. Nessa etapa foram selecionados 25 trabalhos.

Por fim, mais um filtro foi aplicado, dessa vez, analisando a introdução e a conclusão dos trabalhos pré-selecionados até o momento. Essa etapa resultou em 17 trabalhos aceitos. Além desses, foram identificados outros 5 potencialmente relevantes a pesquisa que não haviam sido identificados na busca as bases de dados e foram incluídos de forma manual, totalizando assim, 22 artigos. Vale salientar que os 5 trabalhos incluídos manualmente satisfizeram os critérios de inclusão estabelecidos no protocolo de pesquisa criado. O quadro a seguir traz a lista de trabalhos selecionados.

Quadro 1: Lista de Trabalhos Selecionados

Lista de Trabalhos Selecionados
<i>An Evaluation of the Internal Quality of Business Applications: Does Size Matter?</i> [Li et al., 2011].
<i>Inspection Effectiveness for Different Quality Attributes of Software Requirement Specifications: An Industrial Case Study</i> [Salger et al., 2009].
<i>Assessment of Reusability in Aspect-Oriented Systems using Fuzzy Logic</i> [Nerurkar, 2008].
<i>Quasi-Experiment for Effort and Defect Estimation using Least Square Linear Regression and Function Points</i> [Tenorio et al., 2010].
<i>Issues on Estimating Software Metrics in a Large Software Operation</i> [Barros et al., 2008].
<i>OPI Model: A Methodology for Development Metric based on Outcome Oriented</i> [Thammarak e Intakosum, 2011].
<i>Comparison of Estimation Methods of Costs and Duration in IT Projects</i> [BERLIN 2009].
<i>A simulation approach to six sigma in software development</i> [Bubevski, 2009].
<i>A Metric-Based Multi-Agent System for Software Project Management</i> [Wu et al., 2009].
<i>A Study of Software Metrics</i> [Singh et al., 2011].
<i>Comparing Software Metrics Tools</i> [Linke et al., 2008].
<i>Measuring the Structural Quality of Business Applications</i> [Curtis et al., 2011].
<i>Implementing the Lean concepts of Continuous Improvement and Flow on an Agile Software Development Project - An Industrial Case Study</i> [Swaminathan e Jain, 2011].
<i>Performance Assessment Metrics for Software Testers</i> [Kaniş et al., 2012].
<i>Comparative Study on Applicability of WEBMO in Web Application Cost Estimation within Klang Valley in Malaysia</i> [Hooi et al., 2008].
<i>Generalization of Software Metrics on Software as a Service (SaaS)</i> [Singh et al., 2012].
<i>The 3C Approach for Agile Quality Assurance</i> [Janus e Schmietendorf, 2012].
<i>Knowledge management metrics in software development companies in Brazil</i> [Goldoni e Oliveira, 2009].
<i>Effect of software evolution on software metrics: An open source case study</i> [Johari e Kaur, 2011].
<i>A Survey on Metric of Software Complexity.</i> [Yu e Zhou, 2010]
<i>Cost Drivers of Software Corrective Maintenance: An Empirical Study in Two Companies</i> [Li et al., 2010].
<i>Evaluating Usefulness of Software Metrics - an Industrial Experience Report</i> [Bouwers et al., 2013].

Durante o planejamento da revisão sistemática algumas limitações foram identificadas: a definição do escopo para

essa revisão, devido métricas de *software* ser um assunto de abrangência global, então buscou-se concentrar-se na busca de trabalhos que fossem relacionados a estudos realizados com a indústria.

Devido às limitações das fontes de buscas, artigos relevantes ainda não foram retornados na pesquisa, sendo assim optou-se por incluí-los através de busca manual.

E por fim, a *string* de busca utilizada passou por testes pilotos e após quatro versões, a *string* foi refinada e restrita para o propósito da pesquisa.

A seção a seguir aborda a análise dos resultados obtidos com a aplicação da revisão sistemática da literatura. A análise será feita de forma separada e organizada de acordo com as questões de pesquisa estabelecidas para esse trabalho.

3 Análise dos Resultados Obtidos

Nessa seção, são apresentados os resultados para as questões de pesquisa, parte motivadora desse trabalho. Foram selecionados 22 trabalhos relevantes com a aplicação da revisão sistemática da literatura. Nenhum critério de qualidade (*QA*) é estabelecido em relação aos trabalhos selecionados, pois, talvez os critérios de seleção por si só, já garantam certa qualidade em relação aos estudos. A seguir são apresentadas os resultados por questão de pesquisa estabelecida no protocolo desta pesquisa.

(RQ1) Qual o estado da arte atual em relação estudos sobre métricas de software?

O estado da arte sobre métricas de *software* compreendem trabalhos publicados que abordagem métricas no contexto organizacional de gestão do conhecimento [Goldoni e Oliveira, 2009], Métricas de apoio a sistemas baseados em computação em nuvem e *software as a service* [Singh et al 2012], Métricas para medição de complexidade do *software* [Singh et al. 2012],[Li et al. 2010], métricas no contexto de custos em manutenção corretiva de *software* [Li et al. 2010], Métricas como suporte a mensuração de qualidade em aplicações de negócios [Curtis et al, 2011], Métricas no contexto de inspeção de *software* [Salger et al 2009], Métricas no contexto de qualidade e Testes de *Software* [Kanjij et al 2012], Métricas de *software* de grande escala [Barros et al. 2008], Métricas de desenvolvimento baseadas em resultados [THAMMARAK, e INTAKOSUM, 2011] dentre outros. As seções a seguir ajudarão a contextualizar ainda mais acerca do estado da arte.

(RQ2) Quais as vantagens com o uso de métricas em projetos de software? Existem limitações com essa abordagem?

Buscou-se através dos resultados obtidos com a aplicação da revisão sistemática da literatura, responder essa questão de pesquisa. Para isso, os trabalhos selecionados foram lidos rigorosamente pelos autores deste trabalho.

O trabalho intitulado “*A Study of Software Metrics*” dos autores Gurdev Singh, Dilbag Singh e Vikram Singh, publicado no ano de 2011, abordou tópicos que discutiam as vantagens e limitações identificadas com o uso de

métricas de *software* de forma clara. Além desse, outros trabalhos também abordaram acerca da temática, porém não de forma muito abrangente.

As métricas de *software*, possuem algumas vantagens e desvantagens em relação ao uso. Tais vantagens são apresentadas a seguir [SINGH et al. 2011] [BERLIN et al 2009]:

- Estudo de análise, comparação e crítica de várias linguagens de programação no que diz respeito às suas características;
- Comparar e avaliar as capacidades e produtividade das pessoas envolvidas no desenvolvimento de *software*;
- Elaboração das especificações de qualidade de *software*;
- Na verificação do cumprimento de requisitos de sistemas de software e especificações;
- Inferência sobre o esforço para ser colocado na concepção e desenvolvimento de sistemas de *software*;
- Obtenção de um ideia sobre a complexidade do *software*;
- Tomada de decisões sobre outra divisão do módulo complexo que deve ser feito ou não.
- Prestação de orientação para gerente de recursos para a sua utilização adequada.
- Fornecimento de feedback aos gerentes de software sobre o andamento e qualidade durante as várias fases do ciclo de vida de desenvolvimento de *software*.

Ao mesmo tempo que as métricas de *software* possuem certas vantagens que ajudam no auxílio de algumas atividades dentro do processo de desenvolvimento de *software*, também possuem certas limitações em relação ao seu uso. Singh et al [2011] e Berlin [2009] listam algumas dessas limitações que foram identificadas por eles. São elas:

- 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.

Foi possível identificar que assim como toda e qualquer abordagem, as métricas de *software* também possuem suas vantagens e limitações. Entretanto, apresenta-se

como uma excelente abordagem para o apoio a tomada de decisão através do provimento de informações quantitativas, quando usadas de forma correta.

(RQ3) Quais o tipos de métricas de software identificadas na revisão sistemática da literatura?

De acordo com SINGH et al. [2011; 2012], as métricas de *software* são definidas em dois tipos, a métricas de produto e as métricas de processos. Vários dos trabalhos selecionados abordam alguns tipos de métricas que são usadas em projeto de *software*. Para melhor apresentar os resultados, os autores criaram quadros com informações relevantes como resposta a essa questão de pesquisa.

As informações no quadro estão organizadas da seguinte forma: Nome da métrica, Breve descrição e Referência. Para essa questão de pesquisa foram elaborados três quadros, onde cada um representa um tipo de classificação de métricas: métricas de produto, métricas de produtos que são focadas em orientação à objetos e métricas de processo.

Na coluna referência, os registros que estiverem com * (asterisco) significa que foram inseridos manualmente no processo da revisão sistemática da literatura.

Métricas de Produto

Métricas de Produto, também são conhecidas como as métricas de qualidade e são usadas para medir as propriedades do *software* [SINGH et al. 2012]. Métricas de produtos inclui métricas de funcionalidades, métricas de desempenho, métricas de usabilidade, métricas de custo e tamanho, métricas de complexidade e métricas de estilo. Métricas de Produtos ajudam a melhorar a qualidade dos diferentes componentes e existentes do sistema [SINGH et al. 2011; JOHARI e KAUR, 2011].

Os quadros a seguir mostram de forma sucinta um mapeamento dos tipos de métricas de produto que foram mencionadas nos trabalhos selecionados na revisão sistemática da literatura. O quadro 2 apresenta as métricas de produtos (geral) e o quadro 3 apresenta as métricas de produtos para projetos de orientação a objeto.

Quadro 2: Métricas de Produtos

Métricas de Produto		
Métrica	Descrição	Referência
Métrica de Esforço Base (<i>Effort Base Metric</i>)	Calcula o esforço de trabalho empregado nos projetos de <i>software</i> .	[Barros et al. 2008]
Métrica de Tamanho	Calcula o tamanho do produto de <i>software</i> .	
Métrica de Volume	Métrica de volume pode ser derivada a partir do tamanho de atributo Subsistema.	[Wu et al. 2009]
Métrica de Complexidade	Nesta métrica marca-se gráfico para procurar a complexidade do <i>software</i> . Cada nó representa um bloco de código sequencial e cada aresta corresponde a um percurso criado por uma decisão.	[Singh et al. 2012] [Li et al. 2010]
Métrica de Nro de Variáveis	Essa métrica calcular o número de variáveis definidas na aplicação.	[Li et al. 2010]

Métrica Número de Linhas de Código (NLOC)	Essa métrica faz a contagem da quantidade linhas de código disponíveis na aplicação, excluindo comentários e espaços em branco.	[Singh et al. 2012]
Métrica Número de milhares de instruções fonte entregues (KDSI)	Exprime o tamanho do sistema computacional, dado em milhares de instruções fontes entregues	
Contagem de Pontos de Função (FPC)	Soma ponderada do número de entradas, saídas, consultas e arquivos de dados usados com <i>software</i> .	
Métrica Absoluta	Esta métrica é usada para comparar com um valor fixo proposto <i>software</i> .	
Métrica de Relação	Esta métrica é uma métrica de comparação com respeito a versão anterior do <i>software</i> . Por exemplo, o <i>software</i> proposto é duas vezes mais rápido que o <i>software</i> que acaba de ser concluído.	[Singh et al. 2011]*
Contagem de Tokens	Neste métricas, um programa de computador é considerado uma coleção de símbolos, que podem ser classificadas como operadores ou operandos. Todas as métricas podem ser definida em termos desses símbolos básicos. Esses símbolos são chamados de token.	
Métrica de qualidade de produto	Essa métrica se preocupa com a medição de defeitos do produto	
Número de Defeitos	Métrica que define o número de Defeitos.	[Kanjij et a. 2012]
Severidade do Defeito	Métrica que define a severidade dos defeitos abertos em ciclos de testes.	

Quadro 3: Métricas de produtos OO

Métricas de Produtos (Orientação a Objetos)		
Métrica	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.	
Número de Métodos por classe	Métrica para contagem do número de métodos definidos em uma classe em um programa.	[Johari e Kaur, 2011]* [Singh et al. 2011]*
Número de Atributos	Métrica para contagem do número de atributos de uma classe.	
Métrica de Acoplamento	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 classe dentro da hierarquia herança é o	

Profundidade de árvore de herança (DIT)	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.	

Métricas de Processo

As métricas de processo são conhecidos 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. Métricas de processo ajuda na previsão do tamanho do sistema final e determinar se um projeto sobre a execução de acordo com o cronograma [SINGH et al. 2011].

O quadro a seguir apresenta as métricas de *software* categorizadas como métricas de processo que foram identificadas na revisão sistemática da literatura.

Quadro 4: Métricas de processos

Métricas de Processo		
Nome	Descrição	Referência
Métrica de Reusabilidade	São métricas que medem o nível de reúso nos projetos de <i>software</i> .	[Nerurkar, N.W. 2010]
Métricas de Processos	Esta métrica está preocupado com as aplicações de métodos e ferramentas, utilização de normas, eficácia da gestão e o desempenho do sistema de desenvolvimento.	[Singh et al. 2012]
Métricas de Recursos	Esta métrica está preocupado com a utilização de recursos como esforços, tempo e <i>hardware</i> .	
Métrica de Produtividade	Define estado de produtividade para um subsistema específico durante o processo de desenvolvimento do <i>software</i> .	[Wu et al. 2009]
Métrica Média de ponto de estória por desenvolvidor por dia	Métrica determinada a partir da equação dada abaixo: Taxa de história por iteração = Número total de pontos da história aprovados e fechada pelo cliente na iteração / número atual de dias desenvolvidor trabalhou na iteração.	[Swaminathan, e Jain 2010]

Através dos quadros acima apresentado, foi possível identificar que muitos são os tipos de métricas de *software* disponíveis na literatura específica. Elas vão desde métricas de produtos (com subcategoria para projeto de orientação a objetos) até métricas de processos.

(RQ4) Existem ferramentas que dão suporte ao uso de métricas de software?

Muitos são os tipos de métricas disponíveis na literatura, que abrangem e suportam diversas atividades dentro do processo de desenvolvimento de *software*. Sendo assim, não basta apenas ter as métricas, é preciso saber usá-las. Para isso existem diversas ferramentas que podem ajudar

os gestores de *software* a usar as métricas da forma mais adequada e automatizada possível.

Dentre os 22 trabalhos selecionados, apenas 1 trouxe resultados significativos que pudessem ser usados como subsídio para a contextualização da resposta a essa questão de pesquisa. O trabalho intitulado “*Comparing Software Metrics Tools*” dos autores Rüdiger Lincke, Jonas Lundberg e Welf Löwe da Universidade da Suécia, publicado em 2008, traz um levantamento de algumas ferramentas que são usadas com suporte para o uso de métricas, em especial, uso de métricas de produtos (Orientação a objetos). [LINCKE et al 2008] destaca as seguintes ferramenta: *Anlyst4j*, *CCCC*, *Chidamber & Kemmerers Java Metrics*, *Dependency Finder*, *Eclipse Metrics Plugin 1.3.6*, *Eclipse Metrics 3.4*, *OOMeter*, *Understand for Java* e *VizzAnalyzer*.

Todas as ferramentas apresentadas acima, dão suporte ao uso de métricas como Linhas de Código, Número de Métodos, Número de Filhos, Métricas de Acoplamento entre classes e objetos, Falta de Coesão de Métodos (LCOM), Profundidade de árvore de herança (DIT), Métodos ponderados por classe e Resposta para uma Classe. A seguir serão abordados aspectos e conceito relevantes para a resposta a questão principal dessa pesquisa.

(RQ5) Por que é difícil para a indústria de software realizar estimativas e prover métricas de forma eficiente?

Com base em todas as informações colhidas até o momento, buscou-se identificar possíveis causas ou possíveis características que contribuem com que as atividades de medição e estimativa sejam realizadas de forma ineficiente.

Sabe-se que realizar estimativas e prover métricas de *software* eficientes tem se tornado um grande desafio na área de TI e que essa incapacidade da indústria em estimar *software* com precisão, resulta em derrapagens orçamentais e atrasos nas entregas. Diante disso, questionamentos como: “Por que isso acontece? Por é tão difícil estimar e prover métricas de forma eficiente?” - se mostram relevantes para academia e para a indústria, diante da necessidade de entender esses “porquês”.

Nenhum dos trabalhos selecionados na revisão sistemática da literatura aborda de forma explícita motivos que talvez contribuam para essa dificuldade em relação ao uso de métricas. Entretanto, através de algumas informações contidas nesses trabalhos, pode-se inferir tais motivos.

Em [SINGH et al. 2011], percebe-se através das colocações dos autores que métricas de *software* possuem certa limitações. Com base nessas limitações, os autores deste trabalho inferiram que algumas delas contribuem para o uso ineficiente de métricas em projetos de *software*. Para os autores, as métricas de *software* não são aplicadas de forma eficiente por que são difíceis de aplicar, é um trabalho dispendioso e utiliza em sua maioria dados históricos e empíricos, que são difíceis de verificar e que muitas vezes as empresas não possuem esses tipos e dados, o que resulta na falta de informações sobre projetos anteriores completados.



Já em [Bouwers et al 2013] uma vasta gama de métricas de *software* visando vários níveis de abstração e atributos de qualidade têm sido propostas pela comunidade de pesquisa. Para muitas destas métricas de avaliação consiste em verificar as propriedades matemáticas da métrica, investigar o comportamento da métrica para um certo número de sistemas ou comparando o valor da métrica contra outras métricas quantificando atributos de qualidade relacionados. E isso é um problema quando se pensa em ter métricas eficientes e reais. Dessa forma uma análise estrutural das métricas seria uma solução.

[Bouwers et al 2013] Infelizmente, uma análise estrutural da utilidade de métricas num ambiente de avaliação do mundo real é frequentemente ausente, sendo que essa avaliação é importante para compreender as situações em que uma métrica pode ser aplicada, para identificar áreas de possíveis melhorias, para explorar problemas gerais detectadas pelos indicadores e definir estratégias de solução geralmente aplicáveis. Dessa forma pode-se inferir que a falta de uma análise estrutural da métrica pode contribuir para a geração de métricas ineficientes ou irreais.

Os autores do presente trabalho acreditam que além das possíveis causas apresentadas até o momento, a incapacidade produzir requisitos precisos e a falta de experiência em projetos similares, além da constante evolução da TI, podem contribuir para o uso ineficiente quando se pensa na geração de métricas eficientes e reais. Esse pensamento é confirmado por [BERLIN et al 2009] em seu trabalho intitulado "*Comparison of Estimation Methods of Costs and Duration in IT Projects*" publicado em 2009.

Também acredita-se que a falta de experiência e de conhecimento do profissional contribui significativamente para aumento dessa lacuna da engenharia de *software*. O quadro abaixo mostra um resumo na visão dos autores dos atributos ou limitações que contribuem para a prática ineficiente em relação a geração e uso de métricas de *software*. Esses atributos foram classificados em organizacional/ambiente e técnico.

Quadro 5: Atributos contribuintes para o uso ineficiente de métricas de *software*

Atributos contribuintes para o uso ineficiente de métricas de <i>software</i>	
Organizacional/Ambiente	Técnico
Falta de experiência do profissional	Requisitos imprecisos
Falta de experiência em projetos similares	Falta de análise estrutural
Falta de conhecimento	Utiliza dados históricos que muitas vezes nem existem na organização/projeto.
Constante evolução da TI	São difíceis de aplicar e de verificar

A seção a seguir traz as conclusões e os trabalhos futuros referentes ao estudo realizado.

4 Conclusões e Trabalhos Futuros

Através da revisão sistemática da literatura, foi possível identificar diversos trabalhos relevantes que nos ajudou a identificar possíveis respostas para as questões de pesquisa estabelecidas. Foi constatado que existem diversos estudos sobre métricas realizados com várias aplicações diferentes que vão desde *softwares* convencionais, passam por computação em nuvem, e chegam à gestão do conhecimento.

Esse trabalho ainda mostrou devido o levantamento realizado, os mais diversos tipos de métricas utilizadas e algumas ferramentas de suporte ao uso de métricas de produtos orientados a objetos.

Também ficou evidente que as métricas de *software* proporcionam diversas vantagens para que as utiliza, entretanto deve-se ressaltar que mesmo possibilitando diversas vantagens, essa abordagem oferece algumas limitações.

Foi possível inferir que muitas dessas limitações podem ser fatores que contribuem para que estimativas e métricas sejam feitas de forma ineficiente por quem as usam. Algumas características que foram identificadas compreendem a falta de uma análise estrutural das métricas, a questão de métricas serem difíceis de aplicar pois é um trabalho dispendioso e usam dados históricos e empíricos e que nem sempre estão disponíveis dentro da organização e/ou projeto, além da falta de conhecimento e experiência do usuário e requisitos inconsistentes e/ou imprecisos.

Por fim, pode-se afirmar que vários estudos sobre métricas de *software* foram realizados na indústria, tornando possível identificar e analisar conceitos, características, dentre outras informações a fim de aprimorar os estudos na área e enriquecer o seu conteúdo teórico.

Como sugestão de trabalhos futuros destaca-se a necessidade de ampliar os estudos de forma mais detalhista e realizar experimentos e/ou observações com empresas de TI com o objetivo de identificar fatores que contribuem para o uso ineficiente das atividades de estimativa e medição.

Referências bibliográficas

[BARROS et al 2008] BARROS, Rodrigo C.; RUIZ, Duncan D.; BASGALUPP, Márcio P.; BECKER, Karin. *Issues on Estimating Software Metrics in a Large Software Operation*. 32nd Annual IEEE Software Engineering Workshop, 2008.

[BERLIN 2009] BERLIN, Stanislav. et al, *Comparison of Estimation Methods of Costs and Duration in IT Projects*. Information and Software Technology, vol. 51 pp. 738-748, 2009.

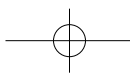
[BIOLCHINI et al. 2005] BIOLCHINI, Jorge; MIAN, Paula Gomes; NATALI, Ana Candida Cruz; TRAVASSOS, Guilherme Horta. *Systematic Review in Software Engineering*. Technical Report. PESC – COPPE/UFRJ, 2005.

[BOUWERS et al 2005] 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.

- [BUBEVSKI, 2009] BUBEVSKI, Voio. *A Simulation Approach to Six Sigma in Software Development*. SCSC '09 Proceedings of the 2009 Summer Computer Simulation Conference, 2009.
- [CURTIS et al, 2011] CURTIS, Bill; SAPPIDI, Jay; SUBRAMAYAM; Jintendra. *An Evaluation of the Internal Quality of Business Applications: Does Size Matter?*. 33rd International Conference on Software Engineering, Honolulu, Hawaii, 2011a.
- [CURTIS et al., 2011] CURTIS, Bill; SAPPIDI, Jay; SUBRAMANYAM, Jitendra. *Measuring the Structural Quality of Business Applications*. 2011 Agile Conference, 2011b.
- [DYBA et al, 2005] DYBA, T.; KAMPENES, V.; SJOBERG, D. *A Systematic Review of Statistical Power in Software Engineering Experiments*. Journal of Information and Software Technology, v. 1, n. 11, 2005.
- [EGOROVA et al., 2010] EGOROVA, Evgenia; TORCHIANO, Marco; MORISIO, Maurizio. *Actual vs. perceived effect of software engineering practices in the Italian industry*. The Journal of Systems and Software, 2010.
- [FENTON e NEIL, 2000] FENTON, Norman E; NEIL, Martin. *Software Metrics: A roadmap*. ICSE '00 Proceedings of the Conference on The Future of Software Engineering, 2000.
- [GOLDONI E OLIVEIRA, 2009] GOLDONI, Vanessa; OLIVEIRA, Mirian. *Knowledge management metrics in software development companies in Brazil*. JOURNAL OF KNOWLEDGE MANAGEMENT, 2009.
- [GOODMAN 1993] GOODMAN, Paul. *Practical Implementation of Software Metrics*, McGraw Hill, London, 1993.
- [HOOI et al., 2008] HOOI, Tan Chin; YUSOFF, Yunus; HASSAN, Zainuddin. *Comparative Study on Applicability of WEBMO in Web Application Cost Estimation within Klang Valley in Malaysia*. IEEE 8th International Conference on Computer and Information Technology Workshops, 2008.
- [JANUS e SCHMIETENDORF, 2012] JANUS, André; SCHMIETENDORF, Andreas. *The 3C Approach for Agile Quality Assurance*. WETSOM 2012.
- [JOHARI e KAUR, 2011] JOHARI, Kalpana; KAUR, Arvinder. *Effect of software evolution on software metrics: An open source case study*. ACM SIGSOFT Software Engineering Notes, 2011.
- [KANIJ et al 2012] KANIJ, Tanjila; MERKEL, Robert; GRUNDY, John. *Performance Assessment Metrics for Software Testers*. CHASE 2012, Zurich, Switzerland, 2012.
- [KITCHENHAM et al. 2004] KITCHENHAM, B. A.; DYBA, T.; JORGENSEN, M. **Evidence-based Software Engineering**. Proceedings of the 26th International Conference on Software Engineering (ICSE'04), p. 273-281, 2004.
- [LI et al. 2010] LI, Jingyue; STALHANE, Tor; KRISTIANSEN, Jan M. W.; CONRADI, Reidar. *Cost Drivers of Software Corrective Maintenance: An Empirical Study in Two Companies*. 26th IEEE International Conference on Software Maintenance in Timișoara, 2010
- [LINCKE et al. 2008] LINCKE, Rudiger, LUNDBERG, Jonas, LOWE Welf. *Comparing Software Metrics Tools*. ISSTA'08, July 20-24, Seattle, Washington, USA, 2008.
- [LITTEL et al., 2008] LITTELL J. H.; CORCORAN J.; PILLAI, V. *Systematic Reviews and Meta-analysis*. USA: Oxford University Press, 2008.
- [MAFRA E TRAVASSOS, 2006] MAFRA, S.N., TRAVASSOS, G.H. *Estudos Primários e Secundários Apoiando a Busca por Evidência em Engenharia de Software, Relatório Técnico ES-687/06*. COPPE/UFRJ. Rio de Janeiro: Brasil, 2006.
- [MOLØKKEN 2003] 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.
- [NERURKAR 2010] NERURKAR, N.W. *Assessment of Reusability in Aspect-Oriented Systems using Fuzzy Logic*. ACM SIGSOFT Software Engineering Notes, 2010.
- [PAI et al 2004] PAI, M.; MCCULLOCH, M.; GORMAN, J.; PAI, N.; ENANORIA, W.; KENNEDY, G.; THARYAN, P.; COLFORD JR., J. *Systematic reviews and meta-analyses: An illustrated step-by-step guide*. The National Medical Journal of India, v. 17, n. 2, p. 86-95, 2004.
- [SALGER et al 2009] SALGER, Frank; ENGELS, Gregor; HOLFMAN, Alexander. *Inspection Effectiveness for Different Quality Attributes of Software Requirement Specifications: An Industrial Case Study*. 2009 ICSE Workshop on Software Quality, 2009.
- [SAMPAIO E MANCINI, 2007] SAMPAIO, R.F; MANCINI, M.C. *Estudos de Revisão Sistemática: Um guia para síntese criteriosa da evidência científica*. Revista Brasileira de Fisioterapia. São Carlos, 2007.
- [SINGH et al. 2011] SINGH, Gurdev; SINGH, Dilbag; SINGH, Vkram. *A Study of Software Metrics*. IJCEM International Journal of Computational Engineering & Management, Vol. 11, January 2011.
- [SINGH et al. 2012] SINGH, Ram; BHAGAT, Avinash; KUMAR, Navdeep. *Generalization of Software Metrics on Software as a Service (SaaS)*. International Conference on Computing Sciences, 2012.
- [SWAMINATHAN, 2012] SWAMINATHAN, Balachander; JAIN, Karuna. *Implementing the Lean concepts of Continuous Improvement and Flow on an*



Agile Software Development Project - An Industrial Case Study. Agile India 2012.

[TENORIO et al., 2010] TENORIO JR, Nelson N; RIBEIRO, Marcelo Blois; RUIZ, Duncan D. *A Quasi-Experiment for Effort and Defect Estimation using Least Square Linear Regression and Function Points.* 32nd Annual IEEE Software Engineering Workshop, 2008.

[YU e ZHOU, 2010] YU, Sheng; ZHOU, Shijie. *A Survey on Metric of Software Complexity.* Information Management and Engineering (ICIME), 2010

[WU et al, 2009] WU, Ching-seh; CHANG, Wei-chun; SETHI, Ishwar, K. *A Metric-Based Multi-Agent System for Software Project Management.* Eighth IEEE/ACIS International Conference on Computer and Information Science, 2009.

