

MEASURESOFGRAM:  
UMA PROPOSTA DE INDICADOR PARA OBSERVAÇÃO  
MULTIDIMENSIONAL DA QUALIDADE DO PRODUTO DE SOFTWARE

Hilmer Rodrigues Neri

Exame de qualificação de doutorado  
apresentado ao Programa de Pós-graduação em  
Engenharia de Sistemas e Computação,  
COPPE, da Universidade Federal do Rio de  
Janeiro, como parte dos requisitos necessários à  
obtenção do título de Doutor em Engenharia de  
Sistemas e Computação.

Orientador: Guilherme Horta Travassos

Rio de Janeiro  
Março de 2019

# SUMÁRIO

<u>1</u>	<u>Introdução</u>	5
<u>1.1</u>	<u>Introdução</u>	5
<u>1.2</u>	<u>Delineamento do problema</u>	7
<u>1.3</u>	<u>Motivação e Questão de Pesquisa</u>	8
<u>1.4</u>	<u>Proposta de Solução</u>	10
<u>1.5</u>	<u>Organização do documento</u>	12
	<u>Referências Bibliográficas</u>	14

# ÍNDICE DE FIGURAS

<a href="#">Figura 1: Esquema conceitual do MeasureSoftGram</a>	<a href="#">11</a>
---	--------------------

---

# ÍNDICE DE TABELAS

[Tabela 1: Estrutura Metodológica desta Pesquisa](#)

12

# 1 Introdução

## 1.1 Introdução

As características pervasivas e ubíquas levam o software a "todo lugar" na sociedade contemporânea. O software deve funcionar em computadores, dispositivos móveis, sistemas embarcados e direciona para uma nova realidade onde não só mais o software se comunica com software, mas coisas se comunicam com coisas. Além disso, a expectativa pela disponibilidade antecipada do produto influenciou a evolução do processo de desenvolvimento no tocante a interatividade da construção e da parcialidade incremental das entregas até atingir a versão final do produto. Diante dessa nova realidade, a atividade de garantia da qualidade de sistemas de software, em especial a qualidade das versões de produtos de software liberadas ao longo do processo de desenvolvimento, tornou-se mais complexa, sendo vital para o sucesso e sobrevivência das organizações que desenvolvem produtos de software.

Ao mesmo tempo, informações geradas por usuários a respeito do uso do software e suas versões estão disponíveis em fóruns de discussão específicos, redes sociais, lojas virtuais e demais repositórios que disponibilizam produtos de software. Essa facilidade relacionada a oferta de informações sobre o produto de software em utilização, aliada as características interativas e incrementais do desenvolvimento de sistemas de software contemporâneos, enseja que engenheiros de software e gerentes de produtos sejam capazes de observar e entender a qualidade do produto de software e suas versões, de forma continuada ao longo do processo de desenvolvimento, acompanhando a sequência de entrega de novas versões do produto que, cada vez mais, são disponibilizadas em intervalos de tempo curtos e com mais frequência para os usuários finais (Fitzgerald e Stol, 2017).

A qualidade do produto de software (QPS) é um objeto de estudo na engenharia de software há aproximadamente quatro décadas. A partir dos modelos seminais de avaliação da qualidade do produto de software, definidos por Boehm et al. (1976) e McCall et al. (1977), os quais identificaram características, fatores, e subcaracterísticas de qualidade hierarquizados e inerentes ao software, os modelos posteriormente elaborados seguiram a inspiração das propostas iniciais de hierarquização destes atributos de qualidade. Essa organização viabilizou a observação isolada de aspectos de qualidade interna, externa e de utilização do produto de software e associados ao fenômeno da qualidade do produto de software, além de contribuir para que modelos mais robustos e recentes estabelecessem conexões lógicas entre as medidas e as descrições das características e

subcaracterísticas (“teorias”) consideradas abstratas e, por conseguinte, de difícil utilização prática por parte da indústria (Mordal-Manet et al., 2009) (Wagner et al., 2012).

Em adição aos trabalhos pioneiros surgiu o modelo proposto por Dromey (1995). Trata-se do primeiro trabalho a estabelecer um mecanismo lógico e sistemático para mapear e relacionar as descrições conceituais de características, subcaracterísticas e fatores, com as definições de estruturas sintáticas presentes nas linguagens de programação. Dromey argumenta que o código-fonte e a atividade de implementação são fontes primárias de observação e percepção das propriedades “concretas” do produto de software, como por exemplo, variáveis e expressões. Orientado pelos trabalhos precursores, o modelo ISO/IEC 9126 (2001) e sua atualização ISO/IEC 25010 (2010) tornaram-se referência conceitual e normativa no que tange a observação da qualidade do produto de software. Adicionalmente, existem várias outras variações dos modelos iniciais, como por exemplo, aqueles que propõem novas características de qualidade, específicas ao contexto de software livre, como (Wasserman, 2006) e SQO-OSS (Samoladas, 2008), além de modelos adicionais, como aqueles encontrados nos resultados do estudo secundário realizado por Miguel et al. (2014).

O conhecimento acumulado ao longo dos tempos torna-se evidente através da observação desses diferentes modelos de avaliação da qualidade do produto: há um vasto conjunto de métricas e medidas identificadas como relacionadas a qualidade do produto de software (Kitchenham, 2010) (Elberzhager et al., 2012) (Tahir e MacDonell, 2012) (Mendonça et al., 2013); dos resultados de estudos sobre os relacionamentos entre diferentes atributos de qualidade (características e subcaracterísticas) que mostram a importância e necessidade de se entender as compensações (*tradeoffs*) associadas, tendo em vista os riscos financeiros com impacto direto no custo do desenvolvimento (Al-Daajeh et al., 2012-b); da utilização de abordagens orientadas à análise de dados-ações, de forma a apoiar a tomada de decisão técnico-gerencial acerca da qualidade do produto de sistemas de software (Zhang et al., 2011) (Abdellatif et al. 2015); no uso de estratégias de experimentação em engenharia de software, em larga escala, para apoiar a escolha de versões de produto de software a serem disponibilizadas para os usuários finais, a partir do comportamento em uso do software (Kohavi et al. 2010) (Lindgren e Münch, 2015) (Schermann et al. 2017) (Fabijan et al., 2017); e por fim, nas diferentes necessidades de informação e especificidades de diferentes contextos, que fazem com que engenheiros de software, gerentes de produto, usuários finais e demais interessados nas informações sobre o

produto tenham que decidir, ao longo do processo de desenvolvimento, em como garantir o alinhamento do desenvolvimento ou a evolução do produto à estratégia de negócio das organizações (Buse e Zimmermann, 2012) (Dybå et al., 2012).

Esse conhecimento, de certa forma, tem sido encapsulado em modelos de QPS recentes, tais como o SQuale (Mordal-Manet et al., 2009), Quamoco (Wagner et al., 2012) e, especialmente, nos modelos QATCH (Siavvas et al., 2017) e Q-Rapids (López et al., 2018), que também tomam por base a ISO/IEC 25010. Entretanto, analisar tão somente aspectos de qualidade interna, externa e de utilização do produto de software isoladamente não é suficiente para que gestores de produtos ou engenheiros de software consigam decidir adequadamente sobre a aceitação ou não de uma versão do produto, ao longo do processo de desenvolvimento ou do produto de software per si, em momento de entrega final.

## **1.2 Delineamento do problema**

A falta de entendimento acerca da qualidade apresentada pelo produto de software ao longo do ciclo de desenvolvimento e em momento de entrega da versão final é muitas vezes a causa de decisões inadequadas de projeto. Aliado a isso, a falta de qualificação dos dados (capturando medidas) previamente as avaliações da qualidade contribui sobremaneira a análises e interpretações equivocadas. Um desafio comum para os profissionais da área de software é a identificação e correção de decisões inadequadas antes que estas possam produzir efeitos indesejáveis no produto de software. Isso provavelmente ocorre porque é mais fácil julgar decisões em engenharia de software após o acontecimento do fato, do que monitorar e avaliar o produto de software, antes que o impacto das decisões na qualidade final do produto seja conhecido (Hoover et al., 2010).

Portanto, a carência de indicadores que permitam apoiar a compreensão ou clareza das características da qualidade do produto de software, principalmente a falta de conhecimento sobre suas relações e influências mútuas, contribui para a tomada de decisões não assertivas (inadequadas ou erradas), comprometendo:

- a) a qualidade do produto de software em desenvolvimento ou em operação;
- b) o comportamento do software em uso; e
- c) a estratégia de negócios das organizações.

Particularmente, a observação da qualidade do produto de software de forma unidimensional (linear), esconde e, em vista disso, dificulta a percepção das relações

entre suas características e subcaracterísticas de qualidade simultaneamente. Portanto, os efeitos positivos e colaterais das relações desses atributos não são conhecidos, representáveis, ou observáveis, afetando a percepção da qualidade do produto de software e influenciando, conseqüentemente, a percepção da qualidade do produto, o que pode afetar as tomadas de decisão ao longo do projeto. Encontrar um indicador multidimensional da qualidade do produto de software pode representar uma alternativa evolucionária na busca de uma solução para uma percepção fidedigna da qualidade do produto de software.

### 1.3 Motivação e Questão de Pesquisa

Analisando os estudos dos modelos de qualidade do produto de software percebemos que: i) há pouca evidência experimental que apoie a análise das relações entre as características de qualidade (Henningsson e Wohlin, 2002) (Al-Daajeh et al., 2012-a); ii) existe dificuldade em agregar estudos sobre os relacionamentos entre as características de qualidade de produto (Svahnberg e Henningsson, 2009); iii) as evidências a respeito das relações entre características de qualidade, em sua maioria, são descritas em alto nível e foram obtidas por meio de *surveys* primários e secundários. Em alguns desses estudos a população é representada por profissionais da prática (Al-Daajeh et al., 2012-b), em outros, por artigos (Haoues et al., 2017); iv) e, principalmente, a observação unidimensional da qualidade do produto de software, ou seja, observar uma ou duas características de qualidade em cada momento, sem considerar seus efeitos colaterais ou influências nas demais características de qualidade do produto. Essas lacunas nos levam a refletir sobre a confiança da representação dos indicadores unidimensionais frente a realidade da percepção da qualidade do produto de software pelos *stakeholders*.

Adicionalmente, estudos experimentais sobre medição e medidas de qualidade de software geralmente relatam a falta de rigor no processo de medição e validação das medidas (Kitchenham, 2010) (Meneely et al., 2013); problemas com escalas entre medidas (Juristo e Moreno, 2010); a dificuldade de estabelecer valores de referência para medidas de software (Lavazza e Morasca, 2016) (Lima et al., 2017); e, por último e não menos relevante, as características do contexto que dificultam, ou mesmo, impossibilitam a comparação de medidas entre diferentes produtos ou projetos, diante das especificidades dos diferentes contextos de projeto (Dybå et al., 2012) (Al-Daajeh et. al, 2012-b).

Ainda que os resultados obtidos sejam de grande valia para a evolução do conhecimento na área e o entendimento da qualidade do produto de software, os



modelos sempre apoiam a observação da qualidade do produto, como apontado anteriormente, sob uma perspectiva unidimensional, focando preferencialmente em características diretamente mensuráveis e internas de qualidade. Entretanto, o referencial de qualidade, se tratado de forma unidimensional, impede uma visão holística da qualidade do produto, limitando a percepção da variação da qualidade relacionada a um atributo em particular, sem permitir a percepção fidedigna da qualidade final do produto, tendo em vista as expectativas dos *stakeholders* para diferentes aspectos da qualidade, conforme será discutido no Capítulo 2. Desta forma, argumentamos que as relações entre as características de qualidade internas, externas e de utilização devem ser simultaneamente consideradas e analisadas por meio de um modelo multidimensional de qualidade, direcionado por uma abordagem orientada à interpretação analítica de medidas, a fim de capturar e representar de forma fidedigna para os *stakeholders* o espectro da qualidade do produto de software. Essa é a principal motivação e diferencial do modelo multidimensional para observação da qualidade do produto de software a ser proposto nesta tese, se comparado aos modelos de qualidade atualmente disponíveis e descritos na literatura técnica.

Tendo em vista que: i) as limitações dos modelos de qualidade atuais afetam a avaliação da qualidade dos produtos de software ao longo do ciclo de desenvolvimento, e que ii) organizações que lidam com o desenvolvimento de sistemas de software contemporâneo necessitam compreender melhor os fatores que afetam a qualidade do produto de software, incluindo suas relações e influências mútuas, para perceber de forma fidedigna a qualidade do produto de software ao longo do ciclo de desenvolvimento iterativo e incremental; a questão de pesquisa norteadora e a ser respondida nesta tese é:

*Como observar a qualidade do produto de software, de forma multidimensional, a partir da combinação das diferentes dimensões e relações da qualidade (interna, externa) do produto ao longo do seu ciclo de desenvolvimento?*

Acreditamos que a resposta a esta pergunta, através da proposição de um indicador multidimensional e evidenciado fidedigno a expectativa da qualidade do produto de software dos *stakeholders*, permita oportunidades de aplicação em diferentes cenários de desenvolvimento, incluindo sua utilização no processo de

tomada de decisão de aceitação de versões, tanto pelas equipes de desenvolvimento quanto pelos gestores de desenvolvimento de produto. Para tanto, entendemos ser relevante atender o seguinte objetivo:

<b>Analisar</b>	características e subcaracterísticas de qualidade de produto de software (internas, externas e uso) representados por atributos de qualidade (fatores, medidas e suas relações)
<b>com o propósito</b>	de caracterizar
<b>com respeito à</b>	Viabilidade e fidedignidade de um indicador multidimensional de qualidade do produto de software combinando diferentes atributos de qualidade
<b>do ponto de vista de</b>	gerentes de projeto e engenheiros de software
<b>no contexto de</b>	projetos de software em ambientes de software-livre e organizações públicas federais brasileiras.

## 1.4 Proposta de Solução

Considerando a questão de pesquisa e os objetivos previamente descritos, apresentamos os conceitos iniciais que subsidiam a proposição de um modelo multidimensional para observação da qualidade do produto de software, o qual denominaremos *MeasureSoftGram*. O nome do modelo intenciona capturar essencialmente a metáfora associada ao eletrocardiograma, onde diferentes indicadores biométricos cardíacos são combinados para representar a situação clínica (qualidade) do coração (Feldman e Goldwasser, 2004).

Desta forma, as contribuições esperadas dessa proposta de tese de doutorado são:

- 1) proposição de um modelo matemático que permite modelar a qualidade de software de forma multivariada e multiespacial, por meio do uso de tensores comparáveis no espaço. Por hora, tensores representam a generalização dos conceitos de vetores e escalares. Logo, um tensor é um *array* multidimensional;
- 2) proposição de um conjunto de regras de transformações matemáticas (normalização, ponderação e agregação), em espaços multidimensionais, construído a partir da aplicação de conceitos da álgebra multilinear e aplicável em para qualquer quantidade de métricas ou medidas;

- 3) um mecanismo, que chamamos de “equalizador da qualidade”, para expor e balancear (*tradeoff*) os relacionamentos entre características e subcaracterísticas da qualidade, além de associar a necessidade de informação a objetivos de medição a serem priorizados;
- 4) um modelo de análise da qualidade construído a partir da aplicação de conceitos trigonométricos, que auxilia a interpretação das transformações multidimensionais e viabiliza a comparação entre os tensores de qualidade, no espaço e a percepção da qualidade no plano;
- 5) Um conjunto de facilidades computacionais e visualizadores gráficos que permitem observar a variação dos indicadores ao longo de versões sucessivas do produto de software.

Na Figura 1 apresentamos a abstração conceitual do MeasureSoftGram. As demais propriedades e detalhes do modelo serão tratados no capítulo 3.

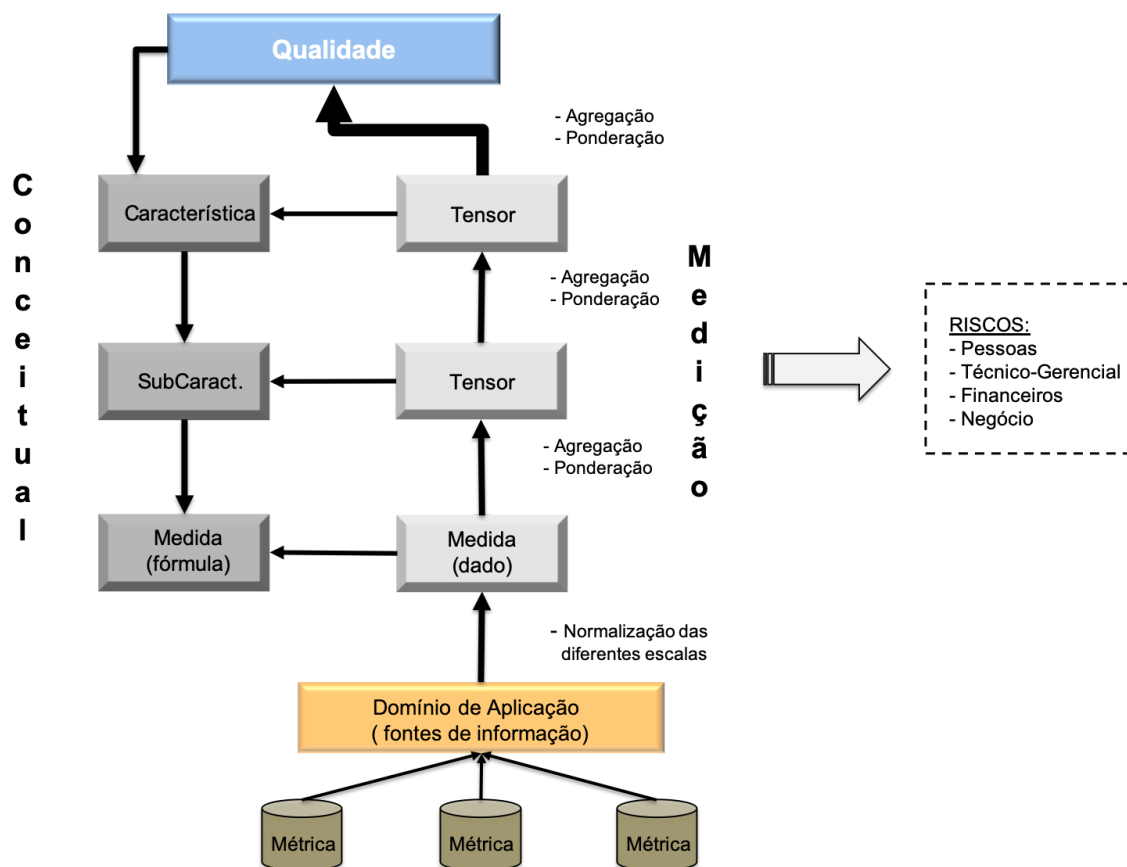


Figura 1: Esquema conceitual do *MeasureSoftGram*

Para a construção do *MeasureSoftGram* seguiremos uma estratégia de experimentação baseada no trabalho de Wohlin e Aurum (2015), que propuseram uma estrutura conceitual para auxiliar pesquisadores a estruturarem o delineamento da pesquisa (*research design*) de seus estudos. Essa estrutura é composta por um conjunto de pontos de decisões organizados em três fases: estratégica; tática e operacional. Na Tabela 1 apresentamos a síntese do delineamento da pesquisa desta proposta de tese, que será detalhada na subseção 3.1 – Metodologia, parte integrante do Capítulo 3.

Fases	Pontos de Decisão	Pontos de Decisão Aplicados
<b>Estratégica</b>	Resultado da Pesquisa	Aplicada
	Lógica da Pesquisa	Indutiva
	Propósito da Pesquisa	Descritiva
	Abordagem da Pesquisa	Crítica e Interpretativista
<b>Tática</b>	Processo da Pesquisa	Mista (Quanti-Quali)
	Método da Pesquisa	Estudo de Caso
<b>Operacional</b>	Método de Coleta de Dados	Observação, Survey e Dados Históricos
	Método de Análise de Dados	Análise Estatística

Tabela 1: Estrutura Metodológica desta Pesquisa

A disponibilização de um indicador multidimensional da qualidade do produto de software, com evidência de viabilidade e fidedignidade quanto a percepção da qualidade fornecida aos *stakeholders* definem o escopo desta tese. Entretanto, acreditamos que sua disponibilidade permitirá apoiar diferentes atividades do processo de desenvolvimento, em destaque: engenharia de software contínua (Fitzgerald e Stol, 2017); experimentação contínua em engenharia de software (Kohavi et al., 2013) (Fabijan et al., 2017) ; e, o uso de uma abordagem analítica para interpretar medidas de qualidade de software que apoiem a tomada de decisão sobre o aceite de versões de produto de software (Zhang et al., 2011) (Bird et al., 2015). Investigar a viabilidade de uso do indicador nestes cenários está fora do escopo deste trabalho, porém compõem relevantes cenários de investigação que merecem atenção em futuro próximo.

## 1.5 Organização do documento

Este trabalho está organizado em cinco capítulos. O primeiro, compreende esta introdução.

No segundo capítulo, apresentamos a revisão bibliográfica das áreas de conhecimento que sustentam as ideias associadas a esta proposta de tese. Essa revisão aborda os estudos dos modelos de qualidade de produto desenvolvidos ao

longo do tempo na engenharia de software, bem como estudos que se propuseram a estudar as relações entre os fatores que governam a qualidade do produto de software. Em seguida, são introduzidos conceitos sobre uma abordagem analítica de interpretação de medidas de software. Também são introduzidos conceitos sobre o processo de desenvolvimento de sistemas contemporâneos e como a qualidade do produto de software vem sendo tratada segundo essa nova realidade. Por fim, são apresentados alguns conceitos matemáticos sobre espaços tensoriais.

Apresentamos no terceiro capítulo a proposta do modelo, onde são detalhados os pontos de decisão referentes à metodologia desta pesquisa. Além disso, são descritos os componentes da solução proposta nesta proposta de tese. Em destaque, a concepção de um modelo matemático e o uso de tensores para modelar e analisar a qualidade do produto de software. São abordadas diferentes questões a serem observadas ao lidar com abstrações no campo vetorial, além do tratamento de dados, como: tratamento de escalas e agregação de medidas; observação das mútuas e múltiplas relações entre variáveis que descrevem um sistema de qualidade de produto de software; ponderação da relevância entre medidas; espaços e subespaços vetoriais usados para representar os fatores da qualidade de produto de software; comparações vetoriais para abstrair e viabilizar a aferição da qualidade de versões de produto de software.

No quarto capítulo, apresentamos os resultados de uma prova de conceito utilizada para indicar a viabilidade inicial do MeasureSoftGram por meio de um protótipo operacional.

Por fim, o quinto capítulo contém as considerações finais, próximos passos planejados e contribuições esperadas desta pesquisa.

## Referências Bibliográficas

- ABDELLATIF, T. M.; CAPRETZ, L. F.; HO, D. Software Analytics to Software Practice: A Systematic Literature Review. Proceedings of the First International Workshop on BIG Data Software Engineering. Anais...: BIGDSE '15. Piscataway, NJ, USA: IEEE Press, 2015. Disponível em: <http://dl.acm.org/citation.cfm?id=2819289.2819300>
- ACAR, E.; Canan, A-B; Haluk, B.; Rasmus, B., Bülent, Y. Multiway analysis of epilepsy tensors. Proceedings 15th International Conference on Intelligent Systems for Molecular Biology {(ISMB)} & 6th European Conference on Computational Biology (ECCB), Vienna, Austria, July 21-25, 2007. Disponível em: <<https://doi.org/10.1093/bioinformatics/btm210>>
- AGRAWAL, R.; GOLSHAN, B.; PAPALEXAKIS, E. A Study of Distinctiveness in Web Results of Two Search Engines. Proceedings of the 24th International Conference on World Wide Web-WWW '15 Companion. New York, NY, USA: ACM, 2015. Disponível em: <<http://doi.acm.org/10.1145/2740908.2743060>>
- ALDAAJEH, S. H.; AL-QUTAISH, R. E.; AL-QIREM, F. A tactic-based framework to evaluate the relationships between the software product quality attributes. International Journal of Software Engineering, v. 5, n. 1, p. 5–26, 2012.
- ALDAAJEH, S. et al. Communing Different Views on Quality Attributes Relationships' Nature. European Journal of Scientific Research, v. 68, n. 1, p. 101–109, 2012.
- AKIYAMA, F. An Example of Software System Debugging. Inform Process Congress. p. 353-359, 1971
- BAKSHY, E.; ECKLES, D.; BERNSTEIN, M. S. Designing and Deploying Online Field Experiments. Proceedings of the 23rd ACM conference on the World Wide Web. Anais...2014
- BARRETO, E. L. Cálculo tensorial. [s.l.] Instituto de Matemática Pura e Aplicada do Conselho Nacional de Pesquisas, 1965.
- BASIL, V. R. A Personal Perspective on the Evolution of Empirical Software Engineering. In: Perspectives on the Future of Software Engineering. [s.l.] Springer Berlin Heidelberg, 2013. p. 255–273.
- BECK, K. et al. Manifesto for Agile Software Development. Manifesto for Agile Software Development, 2001. Disponível em: <<http://www.agilemanifesto.org/>>
- BIRD, C.; MENZIES, T.; ZIMMERMANN, T. The Art and Science of Analyzing Software Data. 1st. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2015.
- BOEHM, B. W. Characteristics of Software Quality. [s.l.] North-Holland, 1978.
- BOEHM, B. W.; BROWN, J. R.; LIPOW, M. Quantitative Evaluation of Software Quality. Proceedings of the 2Nd International Conference on Software Engineering. Anais...: ICSE '76. Los Alamitos, CA, USA: IEEE Computer Society Press, 1976. Disponível em: <http://dl.acm.org/citation.cfm?id=800253.807736>
- BOSCH, J. Building Products as Innovation Experiment Systems. (M. A. Cusumano, B. Iyer, N. Venkatraman, Eds.) Software Business. Anais... Berlin, Heidelberg: Springer Berlin Heidelberg, 2012
- BOSCH, J. et al. The Early Stage Software Startup Development Model: A Framework for Operationalizing Lean Principles in Software Startups. (B. Fitzgerald et al., Eds.) Lean Enterprise Software and Systems. Anais... Berlin, Heidelberg: Springer Berlin Heidelberg, 2013

- BUSE, R. P. L.; ZIMMERMANN, T. Information needs for software development analytics. Proceedings - International Conference on Software Engineering-ICSE. 2012 Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84864194945&partnerID=40&md5=c3501a80b7d9cdd917a665867484ee78>
- BUSE, R. P. L.; ZIMMERMANN, T. Analytics for Software Development. Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research. Anais...: FoSER '10. New York, NY, USA: ACM, 2010 Disponível em: <http://doi.acm.org/10.1145/1882362.1882379>
- CHOMA, J.; GUERRA, E. M.; DA SILVA, T. S. Patterns for Implementing Software Analytics in Development Teams. Proceedings of the 24th Conference on Pattern Languages of Programs-PLoP '17. USA: The Hillside Group, 2017 Disponível em: <<http://dl.acm.org/citation.cfm?id=3290281.3290308>>
- CZERWONKA, J. et al. CODEMINE: Building a Software Development Data Analytics Platform at Microsoft. [S.l.]: IEEE Computer Society. July 2013
- DE FRANÇA, B. B. N.; JERONIMO JUNIOR, H.; TRAVASSOS, G. H. Characterizing DevOps by Hearing Multiple Voices. Proceedings of the 30th Brazilian Symposium on Software Engineering. Anais...: SBES '16. New York, NY, USA: ACM, 2016 Disponível em: <http://doi.acm.org/10.1145/2973839.2973845>
- DE MENDONÇA, Vinicius Rafael Lobo et al. Static analysis techniques and tools: A systematic mapping study. 8<sup>th</sup> International Conference on Software Engineering Advances-ICSEA, 2013.
- DEBOIS, P. Devops: A software revolution in the making. The Journal of Information Technology Management, v. 24, p. 3–5, 2011.
- DIMITRI N., Kleanthis N. Mokios, Nicholas D. Sidiropoulos, and Alexandros Potamianos. 2010. Batch and adaptive PARAFAC-based blind separation of convolutive speech mixtures. IEEE Transactions on Audio, Speech, and Language Processing, 18, 6 (2010), 1193–1207. DOI: <http://dx.doi.org/10.1109/tasl.2009.2031694>
- DINGSØYR, T. D. T. Agile Project Management: From Self-Managing Teams to Large-Scale Development. 37th IEEE/ACM International Conference on Software Engineering, {ICSE} 2015, Florence, Italy, May 16-24, 2015, Volume 2. Anais...2015 Disponível em: <https://doi.org/10.1109/ICSE.2015.299>
- DROMEY, R. G. (1995). A model for software product quality. IEEE Transactions on Software Engineering, 21(2), 146–162. doi:10.1109/32.345830.
- EASTERBROOK, S. et al. Selecting Empirical Methods for Software Engineering Research. [s.l.] Springer, 2007.
- ELBERZHAGER, Frank; MÜNCH, Jürgen; NHA, Vi Tran Ngoc. A systematic mapping study on the combination of static and dynamic quality assurance techniques. Information and software technology, v. 54, n. 1, p. 1-15, 2012.
- EVANGELIA Pantraki and Constantine Kotropoulos. 2015. Automatic image tagging and recommendation via PARAFAC2. In 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP'15). IEEE, 1–6. DOI:<http://dx.doi.org/10.1109/mlsp.2015.7324363>
- ACAR E., Canan Aykut-Bingol, Haluk Bingol, Rasmus Bro, and Bu 'lent Yener. 2007. Multiway analysis of epilepsy tensors. Bioinformatics 23, 13 (2007), i10–i18. DOI:<http://dx.doi.org/10.1093/bioinformatics/btm210>

- FAGERHOLM, F.; GUINEA, A.S.; Mäenpää, H.; Munch J. The RIGHT Model for Continuous Experimentation. *Journal of Systems and Software*, v. 123, p. 292–305, jan. 2017.
- FABIJAN, A.; Pavel D.; Olsson, H. H.; Bosch, J., The Evolution of Continuous Experimentation in Software Product Development: From Data to a Data-driven Organization at Scale. *Proceedings of the 39th International Conference on Software Engineering. Anais...: ICSE '17*. Piscataway, NJ, USA: IEEE Press, 2017 Disponível em: <https://doi.org/10.1109/ICSE.2017.76>
- FELDMAN, J.; GOLDWASSER, G. P. Eletrocardiograma: recomendações para a sua interpretação. *Sociedade de Cardiologia do Estado do Rio de Janeiro - SOCERJ*, v. 17(4), n. out.-dez, p. 251–256, 2004. (Acessado em 8/04/2019) Disponível em: [http://sociedades.cardiol.br/socerj/revista/2004\\_04/a2004\\_v17\\_n04\\_art03.pdf](http://sociedades.cardiol.br/socerj/revista/2004_04/a2004_v17_n04_art03.pdf)
- FELDERER, M. et al. Industrial Evaluation of the Impact of Quality-driven Release Planning. *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. ESEM*, New York, USA, 2014, Disponível em: <<http://doi.acm.org/10.1145/2652524.2652579>>
- FENTON, N. & B. J. *Software Metrics-A Rigorous and Practical Approach*, Third Edition. Third ed. [s.l: s.n.].
- FITZGERALD, B.; STOL, K.-J. Continuous software engineering: A roadmap and agenda. *The Journal of Systems & Software*, v. 123, n. Complete, p. 176–189, 2017.
- HANSEN, S.; PLANTENGA, T. D.; KOLDA, T. G. Newton-based optimization for Kullback-Leibler nonnegative tensor factorizations. *Optimization Methods and Software*, v. 30, n. 5, p. 1002–1029, 2015.
- HAQUES, M. et al. A guideline for software architecture selection based on ISO 25010 quality related characteristics. *International Journal of System Assurance Engineering and Management*, v. 8, n. 2, p. 886–909, 2017.
- HENNINGSSON, K.; WOHLIN, C. Understanding the Relations Between Software Quality Attributes - A Survey Approach, *12th International Conference on Software Quality*
- HOOVER, C. L.; ROSSO-LLOPART, M.; TARAN, G. *Evaluating Project Decisions: Case Studies in Software Engineering*. [s.l.] Addison-Wesley, 2010.
- INSTITUTE, P. M. *Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK)*. Pennsylvania, 2014.
- ISO/IEC 25010 System and software quality models. International Organization for Standardization and International Electrotechnical Commission [s.l.] 2010
- ISO/IEC 9126 Software engineering -- Product quality. [s.l.] ISO/IEC, 2001.
- ISO/IEC:15939 Software Engineering - Software Measurement Process. International Organization for Standardization and International Electrotechnical Commission. [S.l.]. 2002
- JURISTO, N.; MORENO, A. M. *Basics of Software Engineering Experimentation*. 1st. ed. [s.l.] Springer Publishing Company, Incorporated, 2010.
- KAI-WEI Chang, Wen-tau Yih, and Christopher Meek. 2013. Multi-relational latent semantic analysis. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*. 1602–1612
- KEVIC, K.; Kevic, K.; Murphy, B.; Williams, L.; Beckmann, J.; *Characterizing Experimentation in Continuous Deployment: A Case Study on Bing*. *Proceedings*



of the 39th International Conference on Software Engineering: Software Engineering in Practice Track-ICSE-SEIP '17. Piscataway, NJ, USA: IEEE Press, 2017 Disponível em: <<https://doi.org/10.1109/ICSE-SEIP.2017.19>>

- KITCHENHAM, B. What's up with software metrics?--A preliminary mapping study. *Journal of systems and software*, v. 83, n. 1, p. 37–51, 2010.
- KOHAVI, R. et al. Controlled Experiments on the Web: Survey and Practical Guide. *Data Min. Knowl. Discov.*, v. 18, n. 1, p. 140–181, fev. 2009.
- KOHAVI, R.; LONGBOTHAM, R.; WALKER, T. Online Experiments: Practical Lessons. *IEEE Computer*, v. 43, p. 82–85, 2010.
- KOHAVI, R. et al. Online Controlled Experiments at Large Scale. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining-KDD*, New York, NY, USA: ACM, 2013 Disponível em: <<http://doi.acm.org/10.1145/2487575.2488217>>
- LAVAZZA, L.; MORASCA, S. An Empirical Evaluation of Distribution-based Thresholds for Internal Software Measures. *Proceedings of the The 12th International Conference on Predictive Models and Data Analytics in Software Engineering. Anais...: PROMISE 2016*. New York, NY, USA: ACM, 2016 Disponível em: <http://doi.acm.org/10.1145/2972958.2972965>
- LIMA, P. et al. A Metrics Suite for code annotation assessment. *Journal of Systems and Software*, v. 137, p. 163–183, 2018.
- LINDGREN, E.; MÜNCH, J. Software Development as an Experiment System: A Qualitative Survey on the State of the Practice. *Lecture Notes in Business Information Processing. Anais...2015*
- LÓPEZ, L. et al. Q-Rapids Tool Prototype: Supporting Decision-Makers in Managing Quality in Rapid Software Development. *CAiSE Forum. Anais...: Lecture Notes in Business Information Processing*. Springer, 2018
- LUZ, W. P.; PINTO, G.; BONIFÁCIO, R. Building a Collaborative Culture: A Grounded Theory of Well Succeeded Devops Adoption in Practice. *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement-ESEM*, New York, NY, USA: ACM, 2018 Disponível em: <<http://doi.acm.org/10.1145/3239235.3240299>>
- MARINHEIRO, A.; BERNARDINO, J. OpenBRR Evaluation of an Open Source BI Suite. *Proceedings of the International C\* Conference on Computer Science and Software Engineering. Anais...: C3S2E '13*. New York, NY, USA: ACM, 2013 Disponível em: <http://doi.acm.org/10.1145/2494444.2494463>
- MCCABE, T. J. A Complexity Measure. *IEEE Trans. Softw. Eng.*, v. 2, n. 4, p. 308–320, jul. 1976.
- MCCALL, J.; K. RICHARDS, P.; F. WALTERS, G. Factors in Software Quality. Volume I, II and III. *Concepts and Definitions of Software Quality. US Rome Air Development Center Reports*, US Department of Commerce, USA, p. 168, 1977.
- MENEELY, A.; SMITH, B.; WILLIAMS, L. Validating Software Metrics: A Spectrum of Philosophies. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, v. 21, n. 4, p. 24:1–24:28, fev. 2013.
- MENZIES, T. et al. *Sharing Data and Models in Software Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2015.
- MENZIES, T.; ZIMMERMANN, T. Software Analytics: So What? *{IEEE} Software*, v. 30, n. 4, p. 31–37, 2013.

- MIGUEL, J. P.; MAURICIO, D.; RODRIGUEZ, G. A Review of Software Quality Models for the Evaluation of Software Products. CoRR, v. abs/1412.2977, 2014.
- MORDAL-MANET, K. et al. The squal model - {A} practice-based industrial quality model. ICSM. Anais...{IEEE} Computer Society, 2009
- MORGAN, J.; LIKER, J. K. The Toyota Product Development System: Integrating People, Process, and Technology. [s.l.] Taylor & Francis, 2006.
- NEELY, S.; STOLT, S. Continuous Delivery? Easy! Just Change Everything (Well, Maybe It Is Not That Easy). 2013 Agile Conference, {AGILE} 2013, Nashville, TN, USA, August 5-9, 2013. Anais...2013Disponível em: <https://doi.org/10.1109/AGILE.2013.17>
- OLSSON, H. H.; ALLAHYARI, H.; BOSCH, J. Climbing the “Stairway to Heaven” - A Multiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software. (V. Cortellessa, H. Muccini, O. Demirörs, Eds.)EUROMICRO-SEAA. Anais...IEEE Computer Society, 2012Disponível em: <<http://dblp.uni-trier.de/db/conf/euromicro/euromicro2012.html#OlssonAB12>>POPPENDIECK
- OUHBI, S. et al. Evaluating Software Product Quality: A Systematic Mapping Study. 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement. Anais...out. 2014
- PAGE, L.; Brin, S.; Motwani, R.; Winograd, T. The PageRank Citation Ranking: Bringing Order to the Web. [s.l.] Stanford InfoLab, nov. 1999. Disponível em: <<http://ilpubs.stanford.edu:8090/422/>>.
- PANTRAKI, E.; KOTROPOULOS, C. Automatic image tagging and recommendation via PARAFAC2. 25th {IEEE} International Workshop on Machine Learning for Signal Processing, {MLSP} 2015, Boston, MA, USA, September 17-20, 2015. Disponível em: <<https://doi.org/10.1109/MLSP.2015.7324363>>
- PAPALEXAKIS, E. E.; FALOUTSOS, C.; SIDIROPOULOS, N. D. Tensors for Data Mining and Data Fusion: Models, Applications, and Scalable Algorithms. ACM Trans. Intell. Syst. Technol., v. 8, n. 2, p. 16:1--16:44, out. 2016.
- POPPENDIECK, M.; POPPENDIECK, T. Implementing Lean Software Development: From Concept to Cash (The Addison-Wesley Signature Series). [s.l.] Addison-Wesley Professional, 2006.
- Rakesh Agrawal, Behzad Golshan, and Evangelos Papalexakis. 2015. A study of distinctiveness in web results of two search engines. In 24th International Conference on World Wide Web, Web Science Track. ACM. DOI:<http://dx.doi.org/10.1145/2740908.2743060>
- REPÚBLICA, P. Lei nº 8.666/93, de 21 de Junho de 1993, 1993. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/leis/L8666cons.htm](http://www.planalto.gov.br/ccivil_03/leis/L8666cons.htm)
- SAMANTHA Hansen, Todd Plantenga, and Tamara G. Kolda. 2015. Newton-based optimization for Kullback- Leibler nonnegative tensor factorizations. Optimization Methods and Software 30, 5 (April 2015), 1002– 1029. DOI:<http://dx.doi.org/10.1080/10556788.2015.1009977>
- SAMOLADAS, I. et al. The SQO-OSS quality model: measurement based open source software evaluation. Open source development, communities and quality. Anais...2008

- SCHERMANN, G. et al. We're doing it live: A multi-method empirical study on continuous experimentation. *Information and Software Technology*, v. 99, p. 41–57, jul. 2018.
- SJØBERG, T. D. D. I. K.; CRUZES, D. S. What works for whom, where, when, and why?: on the role of context in empirical software engineering. 2012 {ACM-IEEE} International Symposium on Empirical Software Engineering and Measurement, {ESEM} '12, Lund, Sweden - September 19 - 20, 2012. Anais...2012Disponível em: <http://doi.acm.org/10.1145/2372251.2372256>
- SIAVVAS, M. G.; CHATZIDIMITRIOU, K. C.; SYMEONIDIS, A. L. {QATCH} - An adaptive framework for software product quality assessment. *Expert Syst. Appl.*, v. 86, p. 350–366, 2017.
- SLTI/MP. Instrução Normativa N° 04, Secretaria de Logística de Tecnologia da Informação, Ministério do Planejamento 2014.
- SVAHNBERG, M.; HENNINGSSON, K. Consolidating different views of quality attribute relationships. WoSQ@ICSE. Anais...DBLP:conf/icse/SvahnbergH09: {IEEE} Computer Society, 2009
- TAHIR, A.; MACDONELL, S. G. A systematic mapping study on dynamic metrics and software quality. *Software Maintenance (ICSM)*, 2012 28th IEEE International Conference on. Anais...2012
- TANG, D. et al. Overlapping Experiment Infrastructure: More, Better, Faster Experimentation. *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Anais...: KDD '10. New York, NY, USA: ACM, 2010, Disponível em: <http://doi.acm.org/10.1145/1835804.1835810>
- TCU2471 TRIBUNAL DE CONTAS DA UNIÃO. Acórdão 2.471/2008- Plenário, 2008. Disponível em:[http://portal3.tcu.gov.br/portal/page/portal/TCU/comunidades/tecnologia\\_informacao/boas\\_praticas/TCUacordao2471.pdf](http://portal3.tcu.gov.br/portal/page/portal/TCU/comunidades/tecnologia_informacao/boas_praticas/TCUacordao2471.pdf)
- WOHLIN, C. et al. *Experimentation in Software Engineering*. [s.l.] Springer Publishing Company, Incorporated, 2012.
- WOHLIN, C.; AURUM, A. Towards a Decision-making Structure for Selecting a Research Design in Empirical Software Engineering. **Empirical Softw. Engg.**, v. 20, n. 6, p. 1427–1455, dez. 2015.
- Yilun Wang, Yu Zheng, and Yexiang Xue. 2014. Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14)*. ACM, New York, NY, 25–34. DOI:<http://dx.doi.org/10.1145/2623330.2623656>
- ZHANG, D. et al. Software Analytics As a Learning Case in Practice: Approaches and Experiences. *Proceedings of the International Workshop on Machine Learning Technologies in Software Engineering*. Anais...: MALETS '11.New York, NY, USA: ACM, 2011 Disponível em: <http://doi.acm.org/10.1145/2070821.2070829>
- WANG, Y.; ZHENG, Y.; XUE, Y. Travel Time Estimation of a Path Using Sparse Trajectories. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining-KDD '14*.New York, NY, USA: ACM, 2014 Disponível em: <<http://doi.acm.org/10.1145/2623330.2623656>>
- WAGNER, S. et al. The Quamoco Product Quality Modelling and Assessment Approach. *Proceedings of the 34th International Conference on Software Engineering*. Anais...: ICSE '12.Piscataway, NJ, USA: IEEE Press, 2012Disponível em: <<http://dl.acm.org/citation.cfm?id=2337223.2337372>>

WASSERMAN, A.; PAL, M.; CHAN, C. Business Readiness Rating for Open Source.  
Proceedings of the EFOSS Workshop. Anais...Como, Italy: [s.d.]