**SUMÁRIO**

[1 Introdução 3](#_Toc8216477)

[1.1 Introdução 3](#_Toc8216478)

[1.2 Delineamento do problema 4](#_Toc8216479)

[1.3 Motivação 5](#_Toc8216480)

[1.4 Questão de Pesquisa e Objetivo 6](#_Toc8216481)

[1.5 Organização do documento 9](#_Toc8216482)

[Referências Bibliográficas 11](#_Toc8216483)

**ÍNDICE DE FIGURAS**

[Figura 1: Esquema conceitual do MeasureSoftGram 5](#_Toc8157012)

**ÍNDICE DE TABELAS**

[Tabela 1: Estrututura Metodológica desta Pesquisa 10](#_Toc8216578)

# Introdução

## Introdução

A qualidade do produto de software (QPS) é um objeto de estudo na engenharia de software há aproximadamente quatro décadas. A partir dos seminais modelos de avaliação de 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 inerentes a estes produtos, os modelos seguintes seguiram sua inspiração na proposta inicial de hierarquização destes atributos de qualidade. Essa organização viabilizou a observação do fenômeno da qualidade do produto de software e contribuiu para que modelos mais recentemente propostos 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).

Agrupando aos trabalhos pioneiros a proposta de Dromey (1995), 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. Entretanto, existem várias outras variações dos modelos iniciais, como por exemplo, aqueles que propõem novas características de qualidade, específicas ao contexto de software livre, como o OpenBRR (Wasserman, 2006) e SQO-OSS (Samoladas, 2008). Modelos adicionais podem ser encontrados nos resultados do estudo secundário realizado por Miguel J. et al. (2014).

O conhecimento acumulado ao longo dos tempos torna-se evidente através da observação de diferentes modelos de avaliação da qualidade do produto propostos; de um vasto conjunto de métricas e medidas definidas (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 evidenciam a importância e necessidade de se entender as compensações (*tradeoffs)* associadas, uma vez que implica em 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 tomar decisões de modo a garantir o alinhamento do desenvolvimento ou evolução do produto à estratégia de negócio das organizações (Buse e Zimmermann, 2012) (Dybå et al., 2012).

De certa forma, esse conhecimento é encapsulado em modelos de QPS recentemente propostos, como o SQuale (Mordal-Manet et al., 2009), Quamoco (Wagner et al.,2012) e especialmente no QATCH (Siavvas et al., 2017) e Q-Rapids (López et al., 2018), estes também tomam por base a ISO. Entretanto, analisar tão somente aspectos de qualidade interna, externa e em uso, isoladamente, não é suficiente para que gestores de produtos ou engenheiros de software decidam sobre a aceitação ou não de um release ou do produto de software per si.

## Delineamento do problema

A falta de entendimento acerca dos dados gerados ao longo do ciclo de vida do software é muitas vezes a causa de más decisões. Aliado a isso, a falta de qualificação dos dados previamente às análises contribui sobremaneira a análises e interpretações equivocadas. Um desafio comum para os profissionais da área de software é a identificação e correção de más decisões antes que estas possam produzir efeitos indesejáveis. Isso provavelmente ocorre porque é mais fácil julgar decisões em engenharia de software após o acontecimento do fato, do que monitorar e avaliar, antes que seu impacto seja conhecido (Hoover et al., 2010).

Portanto, a falta de compreensão ou clareza sobre as informações que dizem respeito às 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), o que compromete:

1. a qualidade do produto de software em desenvolvimento ou em operação;
2. o comportamento do software em uso; e
3. a estratégia de negócios das organizações.

Finalmente, 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 são desconhecidos e isso afeta a percepção da qualidade do produto de software, influenciando, consequentemente, a tomada de decisão ao longo do projeto.

## Motivação e Questão de Pesquisa

Analisando os estudos dos modelos de qualidade do produto de software percebemos que: i) há pouca evidência experimental que suporte a análise das relações entre as características de qualidade (Henningsson e Wohlin, 2002) (Al-Daajeh et al., 2012-a); ii) existe uma dificuldade em agregar estudos sobre os relacionamentos entre as características de qualidade de produto (Svahnberg e Henningsson, 2009) ; iii) as evidências acerca das relações entre as características de qualidade, em sua maioria, são descritas em alto nível e foram obtidas por meio de *surveys*. Em alguns desses estudos a população foi de profissionais da indústria, em outros, de artigos (Aldaajeh, 2012-b) (Haoues 2017); iv) e, principalmente, a qualidade de produto de software vem sendo observada de maneira unidimensional, ou seja, observando uma ou duas características de qualidade por tempo, sem considerar os efeitos colaterais ou influências nas demais características. Essas lacunas nos levam a refletir sobre quão fidedigna, de fato, é a representação da realidade desse fenômeno.

Além disso, 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, as características do contexto que dificultam, ou mesmo, impossibilitam a comparação de medidas entre diferentes produtos ou projetos, diante das especificidades dos diferentes contextos (Dybå et al., 2012) (Al-Daajeh et. al, 2012-b).

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

Tendo em vista que: i) o problema identificado afeta a avaliação da qualidade de projetos e produtos de software, e que ii) organizações que lidam com o desenvolvimento de software necessitam compreender melhor os fatores que afetam a qualidade do produto de software, além de suas relações e influências mútuas para apoiar a tomada de decisão ao longo do projeto; a questão de pesquisa norteadora e ser respondida nesta proposta de tese é:

*Como observar a qualidade do produto de software, de forma multidimensional e combinando as diferentes dimensões e relações da qualidade (interna e externa) do produto?*

Acreditamos que a resposta a esta pergunta traga oportunidades de uso em diferentes cenários de desenvolvimento, incluindo sua utilização no processo de tomada de decisão de aceitação de releases, tanto pelas equipes de desenvolvimento quanto pelos gestores de desenvolvimento de produto. Para tanto, definimos o seguinte objetivo:

|  |  |
| --- | --- |
| **Analisar** | características e subcaracterísticas de qualidade de produto de software (desenvolvimento e uso) |
| **com o propósito** | de caracterizar um indicador multidimensional de qualidade do produto de software envolvendo fatores, medidas e suas relações, e combinando as dimensões e relações da qualidade interna e externa |
| **com respeito à** | viabilidade e fidedignidade |
| **do ponto de vista** | gerentes de projeto e engenheiros de software |
| **no contexto de** | projetos de software software-livre e organizações públicas federais brasileiras. |

## Proposta de Solução



Considerando a situação descrita anteriormente, apresentamos 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 clinica (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. Importante ressaltar que esse conjunto de regras é válido para qualquer quantidade de métricas ou medidas. Ou seja, o modelo opera o mesmo conjunto de regras transformações caso possua apenas uma única medida ou, caso possua centenas;
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 muldimensionais e viabiliza a comparação entre os tensores de qualidade, no espaço. Por conseguinte, visa apoiar os tomadores de decisão.
5. Um conjunto de facilidades computacionais e visualizadores gráficos que permitirão 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.

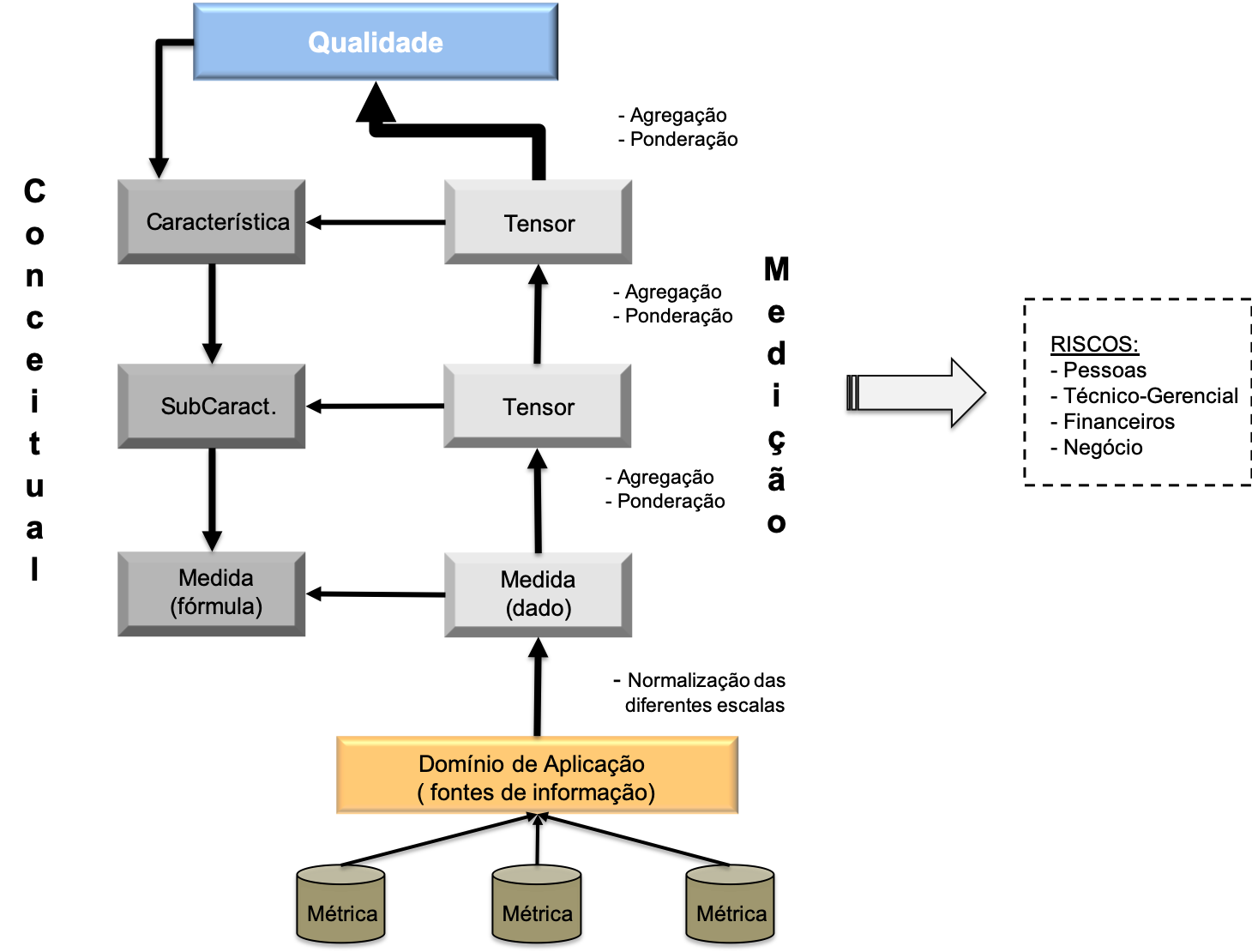


Figura 1: Esquema conceitual do *MeasureSoftGram*

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 desenho 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 desenho da pesquisa desta proposta de tese, que será detalhada na subseção 3.1 – Metodologia, parte integrante do Capítulo 3.

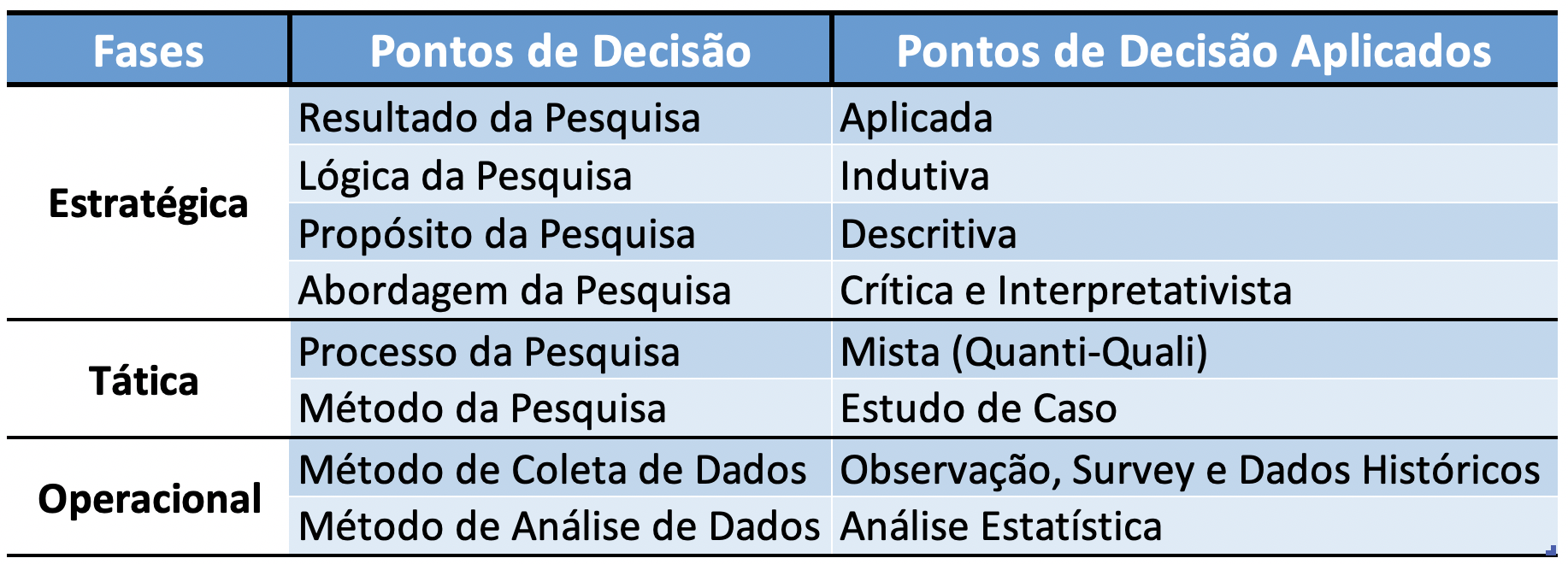


Tabela 1: Estrutura Metodológica desta Pesquisa

## 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 a ideia desta proposta de qualificação. Essa revisão aborda os estudos dos modelos de qualidade de produto propostos 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 de produto. Em seguida, são introduzidos conceitos sobre uma abordagem de apoio a tomada de decisão, em ambientes de desenvolvimento. Também são introduzidos conceitos da engenharia de software contínua, que por sua vez, viabiliza o uso de estratégias de experimentação em larga escala. Por fim, são apresentados alguns conceitos matemáticos sobre espaços tensoriais.

Apresentamos no terceiro capítulo a proposta do modelo, onde são descritos os componentes da solução proposta nesta qualificação. Em destaque, a concepção de um modelo matemático e o uso de tensores para modelar e analisar a qualidade de produto de software de forma multidimensional. 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, além da metodologia desta pesquisa.

No quarto capítulo, apresentamos os resultados de um estudo observacional utilizado para demonstrar a viabilidade 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.

BASILI, 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 DevelopmentManifesto 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, 1976Disponível 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, 2016Disponível em: <http://doi.acm.org/10.1145/2973839.2973845>

DE MENDONCA, Vinicius Rafael Lobo et al. Static analysis techniques and tools: A systematic mapping study. 8th 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...2015Disponível 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>

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.

HAOUES, 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. [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, 2016Disponível 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, 2013Disponível 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 squale 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 Mulitiple-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 na 8.666/93, de 21 de Junho de 1993, 1993. Disponível em: <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 No 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>

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.]