Annotated Bibliography

Review of Software Quality Metrics

1. P. Ward, “Software Quality Metrics Explained With Examples | NanoGlobals,” Oct. 01, 2022. https://nanoglobals.com/glossary/software-quality-metrics/ (accessed Oct. 02, 2022) [1], Rating: (★★★★★)
   d. This article provides an overview of a number of commonly used metrics, along with some examples of how they can be interpreted, quantified, and ultimately used to boost software quality and team productivity.
   This source is included in our research because since this article was published on October 1st, 2022, all references to tools and metrics mentioned therein are current, and the article is written in a clear and concise fashion. **Mohammadali Rahnama was primarily responsible for contributing this source.**

2. M. Agnihotri and A. Chug, “A systematic literature survey of software metrics, code smells and refactoring techniques,” Journal of Information Processing Systems, vol. 16, no. 4, pp. 915–934, 2020 [2], Rating: (★★★★☆)
   d. a. This paper provides an in-depth examination of bad smells in source code as well as applications of refactoring methods to remove those bad smells. Most studies identified or corrected unpleasant smells such as "long method," "feature envy," and "data class." The smells of "feature envy" were detected in 36.66% of the studies that were shortlisted.
   This source is included in our research because an extensive search was conducted in eight digital libraries, and 106 studies were chosen between 2001 and 2019 to analyze refactoring techniques, code smells, and software metrics. **Mohammadali Rahnama was primarily responsible for contributing this source.**

3. L. H. Rosenberg and L. E. Hyatt, “Software Quality Metrics for Object-Oriented Environments,” 2002. [3], Rating: (★★★★☆)
   d. This paper discusses software quality metrics from an object-oriented perspective and focuses on object-oriented concepts that affect software quality. Since this is a study conducted published in 1997, it will prove beneficial to distinguish the difference between past and present software quality metrics. **Alli Said Rashid was primarily responsible for contributing this source.**

4. R. S. Chhillar and S. Gahlot, “An Evolution of Software Metrics: A Review,” in Proceedings of the International Conference on Advances in Image Processing, 2017, pp. 139–143. doi: 10.1145/3133264.3133297. [4], Rating: (★★★★☆)
   d. The research paper examines the various methods of software metrics measurement. Specifically, it provides information on four approaches to capturing software metrics: traditional function-oriented, object-oriented, component-based, and aspect-oriented. Additionally, it discusses the benefits and drawbacks that accompany each technique.

This source is included in our research because it provides a thorough overview of multiple methods to measuring software metrics and presents information such as advantages, disadvantages, and examples for traditional, object-oriented, component-based, and aspect-oriented software metrics. **Alli Said Rashid was primarily responsible for contributing this source.**

5. M. Maddox and S. Walker, “Agile Software Quality Metrics,” in 2021 IEEE MetroCon, 2021, pp. 1–3. [5], Rating: (★★★★☆)
   d. This paper investigates Agile development metrics and proposes a "Core Set" of Agile quality metrics. A concise summary of the findings is provided in the form of metrics drawn from the results and some examples of their use in practice. Additionally, the work will aid (DoD) software development efforts, which have historically been more hesitant to adopt the paradigms of the commercial software industry.
   This source is included in our research because this Agile metric research examined 135 distinct metrics from the literature and industry. This paper concentrated on the metrics that are most useful for assessing the quality of Agile software: test efficiency, defect removal efficiency, production escape percentage, and change failure rate. **Mohammadali Rahnama was primarily responsible for contributing this source.**

6. M. S. Rawat, A. Mittal, and S. K. Dubey, “Survey on impact of software metrics on software quality,” IJACSA) International Journal of Advanced Computer Science and Applications, vol. 3, no. 1, 2012. [6], Rating: (★★★★☆)
   d. This research paper delves into the necessity of software quality metrics in the software development process. It examines various perspectives on software quality and states that quality metrics can assist in controlling the quality of finalized software products. The researchers argue that being able to manage quality through metrics can increase productivity.
   This source is included in our research provides different views on metrics and highlights the necessity of utilizing optimal metrics to obtain ideal results. **Alli Said Rashid was primarily responsible for contributing this source.**

Review of Software Quality Prediction Techniques

7. S. Shafi, S. M. Hassan, A. Arshaq, M. J. Khan, and S. Shamail, “Software quality prediction techniques: A comparative analysis,” in 2008 4th International Conference on Emerging Technologies, 2008, pp. 242–246. doi: 10.1109/ICET.2008.4777508. [7], Rating: (★★★★★)
   d. This research paper elaborates on the significance of software quality prediction techniques. Although there is a plethora of prediction techniques, the paper argues that information on methods to decide which technique is appropriate for specific projects is scarce. An empirical survey was conducted to determine the advantages and disadvantages of thirty techniques and ascertain their results when utilized on various metrics.

This source is included in our research as it provides results to an experiment that demonstrates various software quality prediction techniques, supplying insight on the difference in accuracy of each technique. **Alli Said Rashid was primarily responsible for contributing this source.**

8. F. Alaswad and E. Poovammal, “Software quality prediction using machine learning,” Mater Today Proc, vol. 62, pp. 4714–4720, 2022, doi: https://doi.org/10.1016/j.matpr.2022.03.165. [8], Rating: (★★★★★)
   d. This research paper explores software quality prediction through Machine Learning algorithms. The results favored the machine learning techniques as the most accurate in predicting software quality, allowing researchers to determine their best course of action for contributing to the software quality prediction sector.
   This source is included in our research because it is an up-to-date study and provides meticulous data through organized tables and graphs. **Alli Said Rashid was primarily responsible for contributing this source.**

9. H. Alsolai and M. Roper, “A systematic review of feature selection techniques in software quality prediction,” in 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA), 2019, pp. 1–5. [9] Rating: (★★★★☆)
   d. This research looked at how feature selection methods can be used to foretell software quality. The findings demonstrated that most of the research employed filter selection when choosing features. Defect prediction methods made up over 80% of all software quality prediction methods, while maintainability prediction accounted for only 13%. Research from the PROMISE software project repository and the RELIEF filter selection method was used in 60% of the studies that made the cut.
   This source is included in our research because feature selection techniques are important for improving machine learning models because they improve prediction accuracy while decreasing model creation time. **Mohammadali Rahnama was primarily responsible for contributing this source.’**

10. S. Bouktif, F. Ahmed, I. Khalil, and G. Antoniol, “A novel composite model approach to improve software quality prediction,” Inf Softw Technol, vol. 52, no. 12, pp. 1298–1311, 2010, doi: https://doi.org/10.1016/j.infsof.2010.07.003. [10] Rating: (★★★★☆)
    d. This paper introduces a genetic algorithm to pool the knowledge of multiple models and customize it for a specific software development scenario, producing a "composite" model with improved accuracy. Three different software development scenarios are used to conduct experimental tests of the methodology.
    This source is included in our research because the method proposed in this paper is thought to be both superior to model selection and data combination methods and appropriate for the nature of software data. The paper concludes that learning from pre-existing software models (i.e. software expertise) has

two direct benefits: it helps to avoid the problem of model generalizability and it helps to address the problem of a lack of data in software engineering. **Mohammadali Rahnama was primarily responsible for contributing this source.**

Bibliography

[1] P. Ward, “Software Quality Metrics Explained With Examples | NanoGlobals,” Oct. 01, 2022. https://nanoglobals.com/glossary/software-quality-metrics/ (accessed Oct. 02, 2022).

[2] M. Agnihotri and A. Chug, “A systematic literature survey of software metrics, code smells and refactoring techniques,” Journal of Information Processing Systems, vol. 16, no. 4, pp. 915–934, 2020.

[3] L. H. Rosenberg and L. E. Hyatt, “Software Quality Metrics for Object-Oriented Environments,” 2002.

[4] R. S. Chhillar and S. Gahlot, “An Evolution of Software Metrics: A Review,” in Proceedings of the International Conference on Advances in Image Processing, 2017, pp. 139–143. doi: 10.1145/3133264.3133297.

[5] M. Maddox and S. Walker, “Agile Software Quality Metrics,” in 2021 IEEE MetroCon, 2021, pp. 1–3.

[6] M. S. Rawat, A. Mittal, and S. K. Dubey, “Survey on impact of software metrics on software quality,” IJACSA) International Journal of Advanced Computer Science and Applications, vol. 3, no. 1, 2012.

[7] S. Shafi, S. M. Hassan, A. Arshaq, M. J. Khan, and S. Shamail, “Software quality prediction techniques: A comparative analysis,” in 2008 4th International Conference on Emerging Technologies, 2008, pp. 242–246. doi: 10.1109/ICET.2008.4777508.

[8] F. Alaswad and E. Poovammal, “Software quality prediction using machine learning,” Mater Today Proc, vol. 62, pp. 4714–4720, 2022, doi: https://doi.org/10.1016/j.matpr.2022.03.165.

[9] H. Alsolai and M. Roper, “A systematic review of feature selection techniques in software quality prediction,” in 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA), 2019, pp. 1–5.

[10] S. Bouktif, F. Ahmed, I. Khalil, and G. Antoniol, “A novel composite model approach to improve software quality prediction,” Inf Softw Technol, vol. 52, no. 12, pp. 1298–1311, 2010, doi: https://doi.org/10.1016/j.infsof.2010.07.003.