Analyzing Forty Years of Software Maintenance Models

Valentina Lenarduzzi
Faculty of Computer Science
Free University of Bolzano/Bozen
Bolzano, Italy
valentina.lenarduzzi@unibz.it

Alberto Sillitti
Institute of Information Systems
Innopolis University
Innopolis, Russian Federation
a.sillitti@innopolis.ru

Davide Taibi
Faculty of Computer Science
Free University of Bolzano/Bozen
Bolzano, Italy
davide.taibi @unibz.it

***Abstract*— Software maintenance has dramatically evolved in the last four decades, to cope with the continuously changing software development models and programming languages and adopting increasingly advanced prediction models. In this work, we present the initial results of a Systematic Literature Review (SLR), highlighting the evolution of the metrics and models adopted in the last forty years.**

Keywords-component; Software Maintenance, Systematic Literature Review

I. INTRODUCTION

Software maintenance has been under investigation for more than forty years, with numerous kinds of analysis performed by numerous authors, including those of this paper, and dealing with a variety of process and product aspects [1, 2, 3, 4, 5, 6, 15, 16, 17, 18, 19]. In this work, we introduce the results of a Systematic Literature Review (SLR) depicting the evolution software maintenance models and analyzing the how they have evolved over the past four decades.

The goal of the SLR were to:

- Analyze a chronological overview of the research on maintainability models
- Analyze how maintainability models have evolved
- Understand how the metrics used in maintainability models evolved

We analyzed 1,044 papers written from 1970 to 2015.

Other reviews on software maintenance have been published in recent years [7, 8, 9, 10, 11, 12, 13]. However, our work differs from them in terms of the timeframe, as we consider papers published over 40 years, and comprehensiveness, as we focus on a broad ranges of maintenance models.

As for software maintainability, we consider the ISO/IEC 25010:2011 definition and therefore report on models for analyzability, modifiability, modularity, reusability, and testability.

Methodology

We conducted a Systematic Literature Review [14] followed by a Systematic Snowballing, selecting papers across numerous journals and conferences. Moreover, to avoid excluding any paper, we ran the queries in seven source engines (ACM digital Library, IEEEXplore Digital Library, Science Direct, Scopus, Google Scholar, Citeeser library, Inspec, Springer link, as suggested by Kitchenham [14] [15].

TABLE I Search terms

P	software maintainability	**P terms**	*software maintain, software maintainability, software maintaining, software maintains*
I	model	**I terms**	prediction model, estimation, characterization model, tool model, approach model

Based on the keywords identified in Table I, we adapted the following query for each bibliographic source:

("prediction" OR "estimation" OR "characterization" OR "tool" OR "approach") AND "model" AND ("mainten*" OR "maintain*") AND "software".

We included only papers published in international journals and conferences in the field of Computer Science. We excluded papers that were not peer-reviewed and those not written in English. Moreover, we excluded papers that do not contain any references to maintainability fields in the title and abstract. We also excluded pure maintenance papers in which only a maintenance process was presented without proposing models for assessing and/or predicting maintenance.

We identified 1,044 papers published between 1978 and 2015. We rejected 789 papers after the application of inclusion and exclusion criteria for both title and abstract, and 94 (66 papers for Quality Assessment) after the full read, and we added 10 additional papers from the references, thanks to the systematic snowballing process. Finally, we applied the assessment criteria (see Table II) and ended up with 78 papers that passed the assessment criteria process.

TABLE II Assessment criteria

Assessment criterion	Motivations
Technique adopted	We aim to characterize the different techniques adopted to predict different maintenance characteristics.
Goal	We aim to understand which maintenance characteristics are evaluated by each model. In some cases, the characteristics are related to dependent variables such as effort prediction or bug prediction.
Metrics	To identify also the evolution of the metrics, we considered all metrics or dependent variables in each model.

978-1-5386-1589-8/17 $31.00 © 2017 IEEE
DOI 10.1109/ICSE-C.2017.122

Authorized licensed use limited to: North West University. Downloaded on June 11,2022 at 15:25:46 UTC from IEEE Xplore. Restrictions apply.

II. RESULTS

Unexpectedly, the selected studies related to ISO/IEC 25010:2011 only focus on applicability, with no study predicting modifiability, modularity, reusability, or testability.

Moreover, as a general outcome of the research performed, we identified the following common aspects:

- **Almost no evolution of models:** Almost all the studies built their models from scratch and did not extend existing ones. Every problem was considered as a new one and the model building process only took inspiration from the state of the art, reusing mathematical approaches used by others and adapting some aspects but nothing else.
- **Limited comparison with the literature:** The performance of the developed models is difficult to compare due to the specificity of the problems considered. Therefore, the comparisons are very limited.
- **No third-party validation:** No model was validated independently. All the models were only validated by the proposers, therefore there is no further confirmation regarding the effectiveness of the developed approaches.
- **Nearly impossible replicability of the studies:** Almost all papers are based on private data sets and/or used custom tools that are not available to other researchers to support the replication of the studies.

Such limitations greatly affect the evolution of the field, resulting in continuous development of new and very specific models that fit just some niches and lack generalization, which could help in the creation of more advanced and refined models.

Our study considers different aspects of maintainability, but all of them present a common set of characteristics:

- **Continuous increase in the number of papers:** Maintenance is an increasingly popular research topic, with an increasing number of models and approaches being proposed. Moreover, the number of models proposed is increasing rather than consolidating, highlighting the fact that more research effort is needed in the area to identify really reusable and tunable models that can be applied in different contexts.
- **Continuous increase of the complexity of the models:** The mathematical techniques adopted to analyze the data and build models have been increasing continuously, starting from basic statistical descriptions to advanced data mining and neural network approaches. However, even if such advanced techniques are used, the adaptability of the developed models is very limited.
- **Increase in the size of the data analyzed:** The size of the projects analyzed is also increasing due to the availability of an enormous open source code base that can be easily used to build maintenance models.
- **Limited evolution of the metrics adopted:** Despite the increase in the number of papers in this field, in the amount of analyzed data, and in the complexity of the models, the metrics used to build such models are almost always the same as those developed in the 1970s and 1980s.

Due to all these facts, the industrial applicability of maintenance models is very limited and requires expertise that goes beyond the skills of regular software developers. Therefore, a lot of research is still required to design really reusable and easy-to-tune models that any software developer can easily adopt in everyday software projects affected by a chronic lack of time and budget.

Clearly, the lack of replicability and adaptability of the developed models to different contexts prevents their usage by practitioners in everyday projects due to the advanced skills needed to develop and tune models. However, a deeper investigation of the usage of such models in practice and an analysis of the actual barriers to their adoption are needed to provide input to researchers regarding the development of novel approaches for the definition of maintenance models.

III. THREATS TO VALIDITY

We made every attempt to mitigate threats to validity; here, we highlight the most relevant ones:

Identification of the papers: We searched the selected keywords only in the titles and in the abstracts of the papers. We think that it is quite unlikely that this resulted in not finding papers that deal with the topic but do not include such keywords in the title or abstract.

Data sources: Even though we considered the most popular scientific repositories, we are aware that not all papers might be stored in such repositories; therefore, we also considered Google Scholar, which is capable of aggregating papers from many additional data sources.

Paper filtering: The papers to be included in the study were selected manually, starting with the automatically retrieved ones. To reduce possible biases, the selection was performed by two of the authors following all the steps described in Section II. Inconsistencies were managed via discussions, which always produced final agreement.

Paper analysis: The detailed analysis of the papers was also performed manually by reading the full text. Again, two authors read the papers independently and discussed any inconsistencies. In all of these cases, agreement was reached as well.

REFERENCES

[1] R. Moser, W. Pedrycz, A. Sillitti, and G. Succi, "A model to identify refactoring effort during maintenance by mining source code repositories", 9th International Conference on Product Focused Software Process Improvement (PROFES 2008), Frascati (Rome), Italy, 23 - 25 June 2008.

Authorized licensed use limited to: North West University. Downloaded on June 11,2022 at 15:25:46 UTC from IEEE Xplore. Restrictions apply.

[2] E. Di Bella, I. Fronza, N. Phaphoom, A. Sillitti, G. Succi, and J. Vlasenko, "Pair Programming and Software Defects – a large, industrial case study", Transaction on Software Engineering, IEEE, Vol. 39, No. 7, pp. 930 - 953, July 2013.

[3] P. Abrahamsson, R Moser, W. Pedrycz, A. Sillitti, and G. Succi, "Effort Prediction in Iterative Software Development Processes – Incremental Versus Global Prediction Models", 1st International Symposium on Empirical Software Engineering and Measurement (ESEM 2007), Madrid, Spain, 20 - 21 September 2007.

[4] R. Moser, P. Abrahamsson, W. Pedrycz, A. Sillitti, and G. Succi, "A case study on the impact of refactoring on quality and productivity in an agile team", 2nd IFIP Central and East European Conference on Software Engineering Techniques (CEE-SET 2007), Poznań, Poland, 10 - 12 October 2007.

[5] R. Moser, A. Sillitti, P. Abrahamsson, and G. Succi, "Does refactoring improve reusability?", 9th International Conference on Software Reuse (ICSR-9), Turin, Italy, 11 - 15 June 2006.

[6] P. Ciancarini, and A. Sillitti, "A Model for Predicting Bug Fixes in Open Source Operating Systems: an Empirical Study", 28th International Conference on Software Engineering and Knowledge Engineering (SEKE 2016), Redwood City, San Francisco Bay, CA, USA, 1 - 3 July 2016.

[7] N.F. Schneidewind, "The State of Software Maintenance," IEEE Transactions on Software Engineering, vol. SE-13(3), March 1987, pp. 303-310 doi: 10.1109/TSE.1987.233161.

[8] D.P. Hale and D.A. Haworth, "Software maintenance: a profile of past empirical research," Proceedings of the Conference on Software Maintenance, Oct. 1988, pp. 236-240, doi: 10.1109/ICSM.1988.10168.

[9] D.P. Hale, D.A. Haworth and S. Sharpe, "Empirical software maintenance studies during the 1980s," Proceedings of the Conference on Software Maintenance, Nov. 1990, pp. 118-123, doi: 10.1109/ICSM.1990.131340.

[10] S. Dekleva, "Delphi study of software maintenance problems," Proceedings of the Conference on Software Maintenance, Nov. 1992, pp. 10-17, doi: 10.1109/ICSM.1992.242564.

[11] M. Riaz, E. Mendes, and E. Tempero, "A Systematic Review of Software Maintainability Prediction and Metrics," Proc. 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM'09), 2009, IEEE Press, pp. 367-377, doi:10.1109/ESEM.2009.5314233

[12] R. Abílio, P. Teles, H. Costa and E. Figueiredo, "A systematic review of contemporary metrics for software maintainability," Sixth Brazilian Symposium on Software Components Architectures and Reuse (SBCARS), Sept. 2012, pp. 130-139, doi: 10.1109/SBCARS.2012.15.

[13] A.O. Bajeh., S. Basri and L.T. Jung, "Object-oriented software maintainability measurement in the past decade," International Conference on Advanced Computer Science Applications and Technologies (ACSAT), Nov. 2012, pp. 257-262, doi: 10.1109/ACSAT.2012.54.

[14] Kitchenham, and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software Engineering," EBSE Technical Report EBSE-2007-01.

[15] P. Brereton, B. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," Journal of System and Software, vol. 80(4), 2007, pp. 571-583, doi: 10.1016/j.jss.2006.07.009.

[16] D. Tosi, Lavazza, L. , Morasca, S. , and Taibi, D. , "On the Definition of Dynamic Software Measures", in Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, New York, NY, USA, 2012.

[17] D. Taibi, Lavazza, L. , Morasca, S. , and Tosi, D. , "An Empirical Investigation of Perceived Reliability of Open Source Java Programs", in SAC '12 Proceedings of the 27th Annual ACM Symposium on Applied Computing, Riva del Garda (Trento), Italy March 26-30, 2012, 2012, pp. 1109-1114 .

[18] Lenarduzzi, Lunesu, I. , Matta, M. , and Taibi, D. , "Functional Size Measures and Effort Estimation in Agile Development: a Replicated Study", in XP2015, 2015.

[19] V. Lenarduzzi, Morasca, S. , and Taibi, D. , "Estimating Software Development Effort Based on Phases", in 39th Euromicro Conference on Software Engineering and Advanced Applications , 2014.

Authorized licensed use limited to: North West University. Downloaded on June 11,2022 at 15:25:46 UTC from IEEE Xplore. Restrictions apply.