Predicting Maintainability for Relational Database-Driven Software Applications

Mehwish Riaz
(PhD Enrolment Date: 01-May-2008)

Department of Computer Science, The University of Auckland,
Private Bag 92019, Auckland, New Zealand
mria007@aucklanduni.ac.nz
(+64 9) 373 7599

ABSTRACT

Software maintainability is an important software quality attribute and represents the ease with which a software system can be maintained. Software maintainability prediction models give an estimate of software maintainability and can help software organizations to manage their software and resources effectively. The focus of this research is on the maintainability prediction for relational database-driven software applications. This paper presents the results of a systematic literature review conducted to collect evidence on software maintainability prediction and metrics, the scope of the future research and the methodology intended to be followed.

Keywords: Maintainability, Prediction, Database-driven, Predictors, Metrics.

1. INTRODUCTION

Software maintainability, the ease with which a software system can be maintained [15], is an important software quality attribute and has long been a concern of the software industry [4]. Intrinsically associated with maintainability, is the maintenance process, which has long been known to represent the majority of the costs of the Software Development Life-Cycle (SDLC) [2]. Therefore, the maintainability of a software system can significantly impact software costs. In order to be able to cut down these costs, it is important to create maintainable software. This means that it is important to be able to forecast a software system's maintainability so to effectively manage costs. If software's maintainability can be estimated in advance, it can result in creating maintainable software by addressing the areas in the software that can pose a threat to further maintenance. Research into software maintainability prediction includes proposing and validating maintainability predictors based on measurable factors that have a bearing on the software maintenance activity [2]. The focus of this research is on the maintainability prediction for relational database-driven software applications as the database-driven applications, in my opinion, are perhaps the most common type of applications used in the present time and the choice of relational database is made over other persistency mechanisms like xml, flat files and such. The rationale for making this choice is my belief that this can help a broader practitioner and research community in comparison to other choices for applications and/or persistency mechanisms.

1.1 Prediction in Software Engineering

Prediction, commonly known as estimation [16], is an important part of project planning [8]. It gives an early insight into the software process and helps in creating good quality products within time and budget [21]. Estimates can be made for projects as well as products. When these are made for projects, these are called effort estimates and the process is called effort estimation [22] or software cost estimation [16]. When these are made for maintenance projects, these are called maintenance cost prediction [14] or maintenance project effort estimation [1]. Estimates when made for quality attributes, give a measurable value of the quality of the measured attribute that a software product possesses. The focus of this research is on the prediction of the quality attribute of maintainability.

A software maintainability prediction model enables organizations to predict the maintainability of their software systems, thus providing a means to better manage their maintenance resources in addition to adopting a defensive design [23]. This, in turn, can help in reducing the maintenance effort and the overall cost of a software project.

1.2 Maintainability vs. Maintenance

In order to differentiate between maintenance and maintainability, we consider the following definitions:

Software maintenance: "the process of modifying a software system or component after delivery to correct faults, improve performances or other attributes, or adapt to a changed environment" [15].

Software maintainability: "the ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment" [15].

From the definitions, it is clear that maintenance is the process which is performed as part of the SDLC whereas maintainability is the quality attribute associated with the software product. These are two inherently different but interlocked concepts. Maintenance vs. maintainability is actually process vs. quality attribute of the product, and their estimation is process cost estimation vs. product quality attribute measurement.

1.3 Database-Driven Applications and Their Maintainability

A database-driven application consists of a database, a database management system (DBMS), and a set of applications that interact with the database through the management system [17]. These applications may be written either by using the development environment provided by the DBMS or by any general purpose programming language.

Database-driven applications have gained substantial importance in the modern ways of software development but there is little in the literature on their testing and maintenance [9], [10]. They generally undergo maintenance not only because of software evolution but also due to database evolution when the software requirements change [9], requiring additional information and relationships to be stored [19].

These changes increase schema complexity and result in increased coupling and dependency between database schema and the application [19]. It is generally believed that the more the database application's code is tightly coupled to the database schema, the greater it is affected by the changes in the database schema [11]. Also, features such as Structured Query Language (SQL), exception programming, integrity constraints, and table triggers pose difficulties for maintenance activities [6]. To summarize, the overall maintainability of a database-driven application is impacted by the front end application, the database schema, and the combination of the front end application and the database schema.

The thesis of this research is that predicting maintainability for relational database-driven applications is inherently different to such prediction for other kinds of applications. The goal is to propose maintainability prediction models specific to relational database-driven software applications.

Given how an improved understanding of software maintainability can help organizations create better systems, it is important to understand this process. Therefore, we conducted a Systematic Review (SR) of software maintainability prediction and metrics in order to understand its current state of the art [25]. With the help of the SR we found the gaps in the area based on which we defined the scope and methodology of the future research.

The remainder of the paper is organized as follows: Section 2 briefly describes the SR, its findings, and the findings from related literature. The scope of the future research is discussed in Section 3. A discussion of the methodology and approach of the proposed research is presented in Section 4 followed by conclusions in Section 5.

2. SYSTEMATIC REVIEW

In order to investigate the state-of-art in the area of software maintainability prediction and metrics; and to identify opportunities for future research, a Systematic Review (SR) was conducted by following Kitchenham's guidelines [18]. While the focus of the research was relational database-driven applications, the scope of the SR was broadened to include all types of software applications to ensure that all the relevant information on maintainability prediction techniques, models, and metrics was gathered.

Due to space constraints, the SR process, the research questions considered for the SR, and the findings of the SR could not be included in this paper. Interested readers are referred to Riaz et. al [25] for details. However, it is worth mentioning that the total number of studies selected for the SR was 15. A brief summary of the results is also presented below to better place the context of this research.

2.1 Discussion of SR Results

The results of the SR [25] suggest no obvious choices for maintainability prediction models. Of the models proposed in the literature, few are supported with accuracy measures, few use any form of cross-validation, and few have evidence for external validity. The kinds of models proposed are also quite limited. Out of the 15 selected studies, 12 proposed models for software maintainability prediction and of these only 6 used accuracy measures and 4 mentioned the use of cross-validation techniques for measuring prediction accuracy. Of the 12 studies that have proposed models for software maintainability prediction, 8 have used algorithmic and more specifically regression analysis techniques. The models proposed in 6 studies were specific to the datasets they were based upon, limiting their external validity. It was also observed from the SR results that 2/12 studies using prediction models have compared algorithmic techniques with other techniques – Bayesian Networks and Artificial Neural Networks – where the dataset used comprised of only two data points suggesting the results may not be generalized.

It is, therefore, concluded that further work is needed to (1) develop models for software maintainability prediction having a certain level of external validity and (2) to compare different techniques such as those used for software effort estimation e.g., expert opinion, different algorithmic and artificial intelligence techniques [20] for the prediction of maintainability in software applications.

The SR results [25] also highlight that the prediction techniques employed by the selected studies may be equally applied to any sub-characteristic of maintainability or any type of maintenance. This is due to the reason that except one study which considered the sub-characteristics of understandability, modifiability, and analyzability, none of the studies considered maintainability sub-characteristics or types of maintenance separately.

The SR results [25] also suggest that metrics related to application size, complexity and coupling were the most commonly used software maintainability predictors. The most commonly used measure of maintainability was ordinal scale metric based on expert opinion which, in addition to suggesting the importance of intuitive evaluation of software quality attributes, highlights the fact that metrics using more robust scale types have not been employed often in the literature for quantifying maintainability. Therefore, there is also scope for investigating maintainability metrics that are based on robust scales.

Further analysis of the studies selected in the SR [25] revealed that only 2 studies have each used a relational database-driven application for the dataset and experiment, respectively. Neither of these studies provided any prediction model and only one of these provided metrics that are reported to successfully predict maintainability; these metrics are design level metrics. None of these studies provided any evidence that may be specifically related to maintainability prediction in relational database-driven applications or provided any information on what impacts maintainability of such applications.

2.2 Review of Literature in the Related Areas

The review of literature in the area of database schema changes suggests that there is evidence on topics looking at maintainability at the database schema level such as schema evolution [5] [7], schema versioning [7] [13], metrics for conceptual schema evolution [27], and automatic schema matching [24]. There is also discussion on testing of database-driven applications [3] [17] [6] [10] and evaluation of maintainability of persistency techniques [12]. Some research has been carried out to look at the maintenance of database applications [9] and impact analysis and prediction of database schema changes [19]. Although studies [9] and [19] do not provide specific maintainability metrics or discuss maintainability prediction, they do provide a motivation for exploring metrics and models for maintainability prediction in database-driven applications.

The literature suggests that the aspects of database and software maintainability are considered in isolation. It is either the database-centric concepts or software code that is under study, and the software maintainability for database-driven applications is ignored. Our SR suggests that there is no evidence on maintainability prediction models, metrics, and predictors which might be related to relational database-driven software applications, thus presenting a very clear and wide gap in the area of software maintainability.

3. SCOPE OF FUTURE RESEARCH

The purpose of this research is to propose maintainability prediction model(s) for relational database-driven software applications after identifying the factors that impact their maintainability. The following tasks will be undertaken as a part of the proposed research:

3.1 Evidence Collection on Maintainability Metrics and Predictors for Relational Database-Driven Applications

The purpose of this task (Task 1) is to identify potential maintainability measures to be used as dependent variables and predictors to be used as independent variables in the maintainability prediction models to be created. This involves gathering evidence both through state of art and state of practice. Most of the evidence from the state of art has already been gathered via the SR [25].

Therefore, the main focus of this task is to gather evidence through the state of practice. This evidence is being gathered using interviews with software developers and project managers from software companies that develop relational database-driven software applications and are interested in formalizing the processes they currently employ to estimate maintainability. Our target is to use a sample size of at least 5 to 10 subjects. A larger sample size was not chosen as interviews are generally considered expensive in terms of time. Also, this task is being treated as a pilot and a means to inform the design of the quantitative survey (section 3.3) where a larger sample size would be considered. The interviews are to be carried out in two steps, the first of which is complete.

In order to conduct the interviews, approval from the University of Auckland Human Participants Ethics Committee (UAHPEC) was required and granted. The interviews are recorded on a digital recorder and then transcribed for further analysis.

During the first step of this task, we have interviewed six professionals from the software industry in Pakistan. During the second step of this task, we intend to interview another five professionals from the software industry in New Zealand. We already have conducted two interviews. The purpose of following a two-step approach is to enable better elicitation of factors contributing to maintainability prediction for the said applications and to enable the validation of the results obtained with the help of first step. So far, the types of predictors we have come across in the interviews are those that support the findings from the SR in addition to some soft factors such as use of best practices and process, good documentation, and good design etc.

The interviews in step one were to ask the interviewees about their understanding of 'maintainability'; whether or not they believe if software applications should be maintainable and why; their views and experiences with software maintainability prediction; whether or not their company predicts software maintainability and why; what may be the possible benefits of predicting software maintainability; if their company predicts or measures software maintainability then how it is done and what measures/factors are considered; if their company does not predict or measure software maintainability then what measures/factors would it consider; how is maintainability prediction different for relational database-driven application; and what additional factors should be considered for maintainability prediction and measurement for relational database-driven applications.

The structure of the interviews for second step is the same as that of step one. However, this step is informed by the outcomes of step one for better elicitation of factors for maintainability prediction for the said applications.

3.2 Construction of a Preliminary Theoretical Model

This task (Task 2) will be carried out to propose a preliminary theoretical model that includes maintainability predictors and response for relational database-driven software applications. The information gathered from Task 1 will be interpreted first by identifying the interviewees' responses on the measures and predictors of maintainability for relational database-driven software applications, and then combined with evidence collected from the state of art to present a theoretical model of software maintainability for the said applications. The model would mainly aim to suggest how maintainability for the said applications is related to the predictors identified in task 1 and through the SR. In addition to constructing a theoretical model, this task will also involve classifying the identified predictors that have a bearing on the maintainability of such applications.

3.3 Quantitative Survey to Validate the Theoretical Model

A survey on a relatively larger sample set (Task 3) will be carried out to help validate the proposed theoretical model created in the previous task by providing data that either corroborates or not the findings from the previous step. Here we will aim at a sample size of at least 50 software developers and project managers. A pilot study will be carried out first to validate the survey questionnaire. The responses from the questionnaire will then be analyzed quantitatively which will (1) help validate the theoretical model by comparing the findings of the survey with the theoretical model; and (2) identify the subset of predictors with greatest impact upon maintainability of relational database-driven applications with the help of statistical analysis. This subset will then form the basis for data gathering, to be used in the next phase of the research.

3.4 Construction of Maintainability Prediction Models

This task (Task 4) will involve building maintainability prediction model(s) for relational database-driven software applications using different prediction techniques such as those used for cost estimation including different algorithmic and artificial intelligence techniques etc. [20]. The dependent variable of these models will be a measure of software maintainability and the independent variables will be the predictors identified from Task 3. The models being proposed may be based solely on the data gathered in the previous phase (e.g. multivariate regression), or a combination of data and expert opinion (e.g. Bayesian networks). The primary source being considered for data collection during this task is data on completed relational database-driven software applications developed by some of the companies that participated in Tasks 1 and 2.

The accuracy of the developed prediction models will be determined, using cross-validation method(s), by using various measures of prediction accuracy (e.g. MMRE [20] [26]).

3.5 Comparison of Maintainability Prediction Models

This task (Task 5) will involve a comparison of the different maintainability prediction models for relational database-driven software applications that would be built as part of Task 4. Some of the models built in Task 4 will be models previously suggested in the related literature, used herein to be benchmarked against the new predictions models being proposed. The comparisons will be carried out using a non-parametric test – the Wilcoxon Signed-rank test as it does not make any assumptions regarding the distribution of the data being compared. Based on these comparisons, one maintainability prediction model for relational database-driven software applications will be selected and suggested for use, if only one model is identified. If more than one model are identified as best, suggestions on the conditions under which they give best results will also be provided.

4. METHODOLOGY AND APPROACH

This research falls into the area of empirical software engineering. The methodologies and approach for conducting this research will be evidence-based. The SR [25] conducted was the first step. Further research on this topic will be conducted using both qualitative and quantitative research methods. The interviews being conducted gather qualitative and quantitative data whereas the survey to be conducted would be quantitative in nature and therefore, the responses from the survey will be analyzed statistically.

There are two approaches that may be taken to create the prediction models. The first approach is to collect project data from the software organization(s) against the maintainability predictors suggested by the survey results and then construct the maintainability prediction models. Alternatively, the predictors suggested by the survey results may be used as a basis to build an expert-driven model first, with the participation of software experts from different companies first, and then validated with the help of the project data that would be gathered from the software organization(s) in a later step. The decision on the choice of approach and the specific technique(s) to use for building maintainability prediction models will be based on the results obtained from the survey and on the availability of required data.

During the course of this research, we anticipate to face various challenges and constraints as listed below:

- The quality of the data collected from the interviews and surveys will be largely dependent on the experience, knowledge and skills of the participants; level of honesty in answering the questions; role of the participants in their respective companies; company’s policy of sharing information; and relevance of the questions to the participants’ role, etc. This may pose as a threat to the validity and we are making best effort to choose companies and participants which are the most relevant. For the survey, it will not be possible to have a prior knowledge of the respondents and so, the survey will be designed keeping in mind that all the respondents understand the underlying concepts so that they may provide appropriate answers.
- Data collection for the creation or validation of the maintainability prediction models, depending on which approach is taken, will be a big challenge of this research. It is too early to say if software companies keep the kind of project data we would need and whether or not we would be able to get all or part of the project data required. A possible way to counter this problem is to get all the project data and from then extract the information relevant for this research.

5. CONCLUSIONS

This paper briefly presents the work carried out so far as part of the PhD research of the author; and the scope and methodology of the future research. The work broadly falls in the area of software quality prediction and specifically in the area of software maintainability prediction for relational database-driven software applications. The paper briefly discusses the results of the SR conducted to establish the current state of art in the mentioned area. The paper then discusses the scope of the future work in order to address the gaps found with the help of the SR and the study of other relevant literature. The scope, in brief, includes determining the maintainability metrics and predictors for relational database-driven software applications; the creation of a theoretical model for maintainability prediction of said applications; and the activities of data collection, model(s) creation and model(s) comparison. The methodology and approach for carrying out the future research activities is also presented.

6. REFERENCES

[1] Ahn, Y. et. al. 2003. The Software Maintenance Project Effort Estimation Model Based on Function Points. J Softw Maint Evol, 15, 2, 71 – 85.

[2] Bhatt, P. et. al. 2006. Influencing Factors in Outsourced Softw. Maintenance. ACM SIGSOFT SE Notes, 31, 3 (May 2006), 1 – 6.

[3] Chays, D. and Deng, Y. 2003. Demonstration of AGENDA Tool Set for Testing Relational DB Applications. ICSE’ 03, 802-803.

[4] Coleman, D. et. al. 1994. Using Metrics to Evaluate Software System Maintainability. IEEE Computer (Aug. 1994), 44 – 49.

[5] Curino, C. et. al. 2008. Graceful Database Schema Evolution: the PRISM Workbench. Proc. VLDB Endow. 1, 1 (Aug. ‘08), 761-772.

[6] Daou, B. et. al. 2001. Regression Testing of DB Applications. SAC ‘01.

[7] Franconi, E. et. al. 2000. A Semantic Approach for Schema Evolution and Versioning in Object-Oriented Databases. LNCS, Springer Verlag 1048-1062.

[8] Furulund,K.M. and Moløkken-Østvold,K. 2007. Increasing Software Effort Estimation Accuracy - Using Experience Data, Estimation Models and Checklists. QSIC 2007, 342 – 347.

[9] Gardiokiotis, S. K. et. al. 2004. A Structural Approach towards the Maintenance of Database Applications. IDEAS ‘04.

[10] Gardiokiotis, S. K., and Malevris, N. 2006. Program Analysis and Testing of Database Applications. ACIS-COMSAR ’06.

[11] Gardiokiotis, S. K. et. al. 2007. An Agent-Based Approach for the Maintenance of Database Applications. SERA ‘07.

[12] Goldschmidt, T. et. al. 2008. A Case Study Evaluation of Maintainability and Performance of Persistency Techniques. ICSE’ 08.

[13] Golfarelli, M. et. al. 2005. Schema Versioning in Data Warehouses. LNCS, Springer Berlin / Heidelberg, 415-428.

[14] Granja-Alvarez, J.C. and Barranco-García, M.J. 1997. A Method for Estimating Maintenance Cost in a Software Project: A Case Study. J Softw Maint Evol, 9, 3,161 – 175.

[15] IEEE Std. 610.12-1990. 1993. Standard Glossary of SE Terminology. IEEE Computer Society Press, Los Alamitos, CA.

[16] Jorgensen, M. and Shepperd, M. 2007. A Systematic Review of Software Development Cost Estimation Studies. IEEE T Software Eng, 33, 1 (Jan. 2007), 33 – 53.

[17] Kapfhammer G. M. and Soffa, M. L. A Family of Test Adequacy Criteria for Database-Driven Applications. ACM SIGSOFT Software Engineering Notes, Volume 28, Issue 5 (September 2003). ACM, 2003.

[18] Kitchenham, B. 2007. Guidelines for Performing Systematic Literature Review in SE. EBSE Technical Report, 2.3. Keele University.

[19] Maule, A. et. al. 2008. Impact Analysis of Database Schema Changes. ICSE ’08. 451 – 460.

[20] Mendes, E. and Mosley, N. 2006. Web Engineering. Springer-Verlag Berlin Heidelberg New York. ISBN-13 978-3-540-28196-2.

[21] Mendes, E. and Mosley, N. 2008. Bayesian Network Models for Web Effort Prediction: A Comparative Study, IEEE T Software Eng, 34, 4. pp 723 – 737. IEEE Computer Society.

[22] Mendes, E. 2008. The Use of Bayesian Networks for Web Effort Estimation: Further Investigation. ICWE '08, 203 – 216.

[23] Oman, P., and Hagemeister, J. 1994. Construction and Testing of Polynomials Predicting Software Maintainability. J Syst Software, 24 (1994), 251 – 266.

[24] Rahm, E., and Bernstein, P.A. 2001. A Survey of Approaches to Automatic Schema Matching. The VLDB Journal, Springer Berlin / Heidelberg. 334-350.

[25] Riaz, M. et. al. 2009. A Systematic Review of Software Maintainability Prediction and Metrics. ESEM 2009.

[26] Shepperd, M.J. et. al. 1996. Effort Estimation using Analogy. ICSE ‘96.

[27] Wedemeijer, L. 2001. Defining Metrics for Conceptual Schema Evolution. DB Schema Evolution and Meta-Modeling, 220-244. LNCS.