Electronic Research Archive of Blekinge Institute of Technology
<http://www.bth.se/fou/>

This is an author produced version of a journal paper. The paper has been peer-reviewed but may not include the final publisher proof-corrections or journal pagination.

Citation for the published Journal paper:

Title:

Author:

Journal:

Year:

Vol.

Issue:

Pagination:

URL/DOI to the paper:

Access to the published version may require subscription.

Published with permission from:

International Journal of Software Engineering
and Knowledge Engineering
Vol. 23, No. 2 (2013) 1–16
© World Scientific Publishing Company
DOI: 10.1142/S0218194013006299

MAINTAINABILITY PREDICTORS FOR RELATIONAL DATABASE-DRIVEN SOFTWARE APPLICATIONS: EXTENDED RESULTS FROM A SURVEY

MEHWISH RIAZ*, EWAN TEMPERO† and MUHAMMAD SULAYMAN‡

*Department of Computer Science
The University of Auckland
38 Princes Street, Auckland, New Zealand
*mria007@aucklanduni.ac.nz
†e.tempero@cs.auckland.ac.nz
‡msul028@aucklanduni.ac.nz*

EMILIA MENDES

*College of Computing
Blekinge Institute of Technology, Karlskrona, Sweden
emilia.mendes@bth.se*

Received

Revised

Accepted 9 January 2013

Software maintainability is a very important quality attribute. Its prediction for relational database-driven software applications can help organizations improve the maintainability of these applications. The research presented herein adopts a survey-based approach where a survey was conducted with 40 software professionals aimed at identifying and ranking the important maintainability predictors for relational database-driven software applications. The survey results were analyzed using frequency analysis. The results suggest that maintainability prediction for relational database-driven applications is not the same as that of traditional software applications in terms of the importance of the predictors used for this purpose. The results also provide a baseline for creating maintainability prediction models for relational database-driven software applications.

Keywords: Software maintainability; relational database-driven software applications; survey; predictors; frequency analysis.

1. Introduction

Software maintainability is 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 [6]. It is inherently associated with the process of software maintenance which has long been known to represent a major share of overall

2 *M. Riaz et al.*

1 software costs [12]. In order to manage these costs, it is important to understand,
2 predict and improve software maintainability [12]. Software maintainability pre-
3 diction involves proposing and validating predictors that have a bearing on software
4 maintainability [1] and then employing these predictors to create software main-
5 tainability prediction models [13]. With the help of a maintainability prediction
6 model, software organizations can better manage their maintenance resources and
7 adopt a defensive design [11].

8 Database-driven software applications have gained substantial importance in
9 modern software development [4]. Relational databases are considered to be the
10 most used and successful type of databases for over three decades [15]. Data man-
11 agement has almost become synonymous with Relational Database Management
12 Systems (RDBMS) [15] as most software vendors have adopted the use and con-
13 struction of relational approach to database management [20]. Consequently, rela-
14 tional databases enjoy tremendous success both in Information Technology (IT)
15 shops as well as in data centers [15]. A relational database consists of a collection of
16 tables that are interrelated via referential integrity constraints and whose structure
17 is defined by a schema written using the Structured Query Language's (SQL) data
18 definition constructs [16]. Relational databases and the underlying relational models
19 are considered to be more formalized and inherently different from other types of
20 persistency mechanisms such as file systems [2], network or graph models [17], and
21 object-oriented databases [18].

22 Database-driven applications consist of a database, a Database Management
23 System (DBMS), and a set of applications that interact with the database through
24 this management system [7]. Relational database-driven applications are thus those
25 applications that interact with a relational database through a RDBMS. These
26 applications are often written in object-oriented programming languages and so
27 result in impedance mismatch problem [9]. When a relational database schema
28 changes such as due to adding more information, it leads to increased database
29 schema complexity as well as increased coupling between the database and the
30 software application [9]. The changes sometimes also have an impact on how the data
31 access logic i.e., how applications interact with the relational database, has been
32 implemented [9]. Therefore, the maintainability of relational database-driven
33 applications is impacted by database-specific features, application-specific fea-
34 tures [19], as well as by the interaction of the application with the relational database
35 or implemented data access logic.

36 Given the importance of relational database-driven software applications in
37 modern software development, it is important to investigate the factors that impact
38 upon their maintainability in order to enable maintainability prediction. This paper
39 is a step towards predicting maintainability of relational database-driven application
40 by providing a validated list of predictors that have an impact on the maintainability
41 of relational database-driven applications. This list of predictors was initially gath-
42 ered with the help of twelve interviews conducted with software professionals [13].
43 The interviews' analyses resulted in the identification of 120 predictors. The focus of

the work presented herein is to rank these predictors, with respect to the strength of their impact on the maintainability of relational database-driven applications, via another survey conducted with a larger sample of software professionals. This paper extends the work presented in Riaz *et al.* [21] by supplying further details and insights into the collected evidence. The main contributions of this paper are therefore to:

- Provide evidence that the predictors of maintainability considered important in the context of traditional software applications do not hold the same level of importance in the context of relational database-driven software applications i.e., the predictors relevant to the relational database-end hold more importance in the context of relational database-driven software applications.
- Provide further details on the predictors of maintainability presented by Riaz *et al.* [21] within the context of relational database-driven software applications.
- Provide further details on the different categories that the maintainability predictors for relational database-driven applications belonged.
- Provide further details on the process of ranking the maintainability predictors for relational database-driven software applications in terms of their relative importance to predicting maintainability.

The remainder of the paper is organized as follows. Section 2 gives an account of the related work. Section 3 details the research methodology. Section 4 presents the results. A discussion on the results and threats to validity are given in Sec. 5, followed by conclusions in Sec. 6.

2. Related Work

The research presented herein is informed by the results of a Systematic Review (SR) conducted on the topic of ‘Software maintainability prediction and metrics’ [12] and twelve interviews conducted with software professionals [13] aimed at gathering evidence relating to the maintainability prediction of relational database-driven applications.

The results of the SR revealed very little evidence on maintainability prediction. The total number of studies selected in the SR was 15. These studies were further analyzed to assess if the datasets used in these studies completely or partially comprised relational database-driven software applications. Only three of these studies [3, 5, 8] had used relational database-driven software applications in their datasets for the purpose of their research. However, only one of these studies [5] presented a maintainability prediction model but did not provide its prediction accuracy; the other two studies only presented measures [3] and factors [8] that impacted upon software maintainability, without any associated maintainability prediction model. Further analysis of these studies revealed that although they used relational database-driven applications, none of their proposed predictors or factors related specifically to a back-end database or to the interaction between a back-end database and the front-end application.

To further investigate maintainability prediction in the context of relational database-driven applications, interviews with software professionals were conducted to gather evidence from practice [13]. The interviews were semi-structured in nature and aimed at investigating how the maintainability of relational database-driven applications is understood, measured, and predicted in practice; and what are the predictors of maintainability in the specific context of relational database-driven applications. The results from the interviews revealed that a formal prediction model, or approach to predicting maintainability of relational database-driven applications is not used in practice. The practitioners rely on expert judgement when assessing maintainability of relational database-driven applications. Content analysis was used to analyze the data from interviews. The results from the interviews combined with that of the SR resulted in a list of 120 factors that impact upon the maintainability of relational database-driven applications and may be used as their maintainability predictors. These factors were analyzed keeping in mind how many interviewees mentioned these but did not indicate the relevant importance with regards to predicting maintainability. This paper is an attempt at identifying which of these predictors were most relevant and were more important in the entire set of 120 predictors.

3. Research Methodology

3.1. *Survey-based questionnaire*

In order to determine which maintainability predictors were the most relevant from the list of predictors compiled with the help of SR and interviews' results, a survey was conducted with software practitioners. The list used in the survey comprised a total of 120 predictors [14]. The survey aimed at ranking these predictors in terms of their relative impact on the maintainability of relational database-driven software applications.

The survey questionnaire had 3 parts: The first part is related to the respondents' characteristics i.e., job function at the company, highest qualification, and total experience; and companies' characteristics i.e., year it was established, whether it is involved in developing relational database-driven applications, and the DBMS used most often.

The second part is related to the 120 maintainability predictors which belonged to one of the 10 categories below.

3.1.1. *Database design*

Correct definition of entities; Normalization of the database schema; De-normalization of the database schema according to application needs; Stored procedures; Inline queries; Number of tables; Number of columns in a table; Number of triggers; Number of indexes; Correct use of indexes; Number of SQL queries; Complexity of SQL queries; Number of cursors; Use of sub queries instead of cursors; Use of loops instead of cursors; Unnecessary keys in the database schema; Use of character fields

as keys; Metadata schemas and summary tables; and Creation and maintenance of data dictionary.

3.1.2. *Relational database-driven applications*

Choice of database engine; Choice of data adapters; Choice of data types used in the database; Optimal use of database features; Amount of data the database holds; Number of database calls; Procedures for data transfer between server and clients; Capability of application to interact with various database engines; Easy integration with other applications; Implementation of basic checks at the database level; Use of features provided by database engines; Performance of SQL queries; Physical placement of files on the database; Variety of data storage formats; and Maintenance of database logs.

3.1.3. *Design aspects of relational database-driven applications*

Good software design or architecture, Customizable and parameterized application, Good database design, Complexity of database design, Tiered or layered application (e.g., a separate data access layer), Information coming to a module or object for processing, Information sent out by a software module or object for further processing, Number of modules/objects controlled or connected to a module/objects (from a layered architecture orientation point of view), and Number of modules/objects being controlled or connected by a module/objects (from a layered architecture orientation point of view).

3.1.4. *Size, scope, and complexity of relational database-driven applications*

Complexity of the application and its domain; Scope of the application; Volume of the application; Number of executable files; Lines of code; Number of lines of comments; Number of independent paths through an application; Number of variables used in an application; Scope of the variables used in an application; Number of decision conditions in an application; Methods in a class do not access common or same variables; Classes with objects of other classes as member variables; Difficulty in writing and understanding an application; Number of operators and operand used in an application; Number of overridden methods in an application; Number of private members of a class; Depth of Inheritance Tree in an application; Number of methods and their complexity; Number of methods in a class and invoked by the class; Number of times an object sends a message to a different object; Number of different attributes or member variables of a class; and Number of children of a class.

3.1.5. *Quality, quality assurance, and testing of relational database-driven applications*

Good quality (optimal) code; Strong emphasis on quality assurance; Number of bugs in the application; Application's time to recover from a failure; Availability and

6 *M. Riaz et al.*

1 performance of the application; Data quality; Quality of comments; Complete, up-to-
2 date and high quality documentation; Flexibility and scalability of the application;
3 Strong emphasis on properly conducting the requirements phase; Understandability
4 of the application; Design reviews between the team and with the client; Code in-
5 spection, formal reviews and peer-reviews; Unit testing; Regression testing; Load
6 testing; Testing of all use cases; Scenario-based testing; User acceptance testing; and
7 Reproducibility of behaviour (predictable behaviour when running live).

9 *3.1.6. Change and maintenance of relational database-driven applications*

10 Ability to foresee future changes to the application; Maintaining logs and changes to
11 the application; Formal versioning using a Version Management System; Effort and
12 cost spent on maintenance; Time to perform a maintenance task; Type of mainte-
13 nance task; Understanding a change request and its impact upon maintainability;
14 Changes in the database schema; Number of different versions of the application; and
15 Rate of technological change.

17 *3.1.7. Development and deployment environment of relational* 18 *database-driven applications*

19 Use of packaged software (or reusable components); Variety of processors; Devel-
20 opment and testing tools e.g., development tools, IDEs or frameworks and auto-
21 mated testing tools; Using frameworks for development that allow test harnessing;
22 Knowledge of the application's deployment environment; Number of concurrent
23 users using the application; and Variety of telecommunication protocols.

26 *3.1.8. Web applications with relational database back-end*

27 Number of Web pages or Graphical User Interfaces; Number of Web Objects (e.g.,
28 Java applets, images, multimedia objects like sounds and movies, flash objects, etc.);
29 Number of server side scripts; Number of client side scripts; Number of different
30 programming and scripting languages used to develop an application; Pages or User
31 Interfaces (UIs) linked to other pages or UIs in the application; and Data exchange
32 between different UIs or Web pages.

34 *3.1.9. Organizational culture and policies*

35 Organizational culture of continuous improvement; Following standards (for coding
36 and naming) and practices; Introducing policies for making changes to the applica-
37 tion and/or database; and Organization's data archive strategy.

40 *3.1.10. Team communication and project management*

41 Pair programming; Skilled and experienced application developers, architects,
42 maintainers, and database architects; Communication between front-end developers
43

1 and database architects; Time to market; Cost of the project; Marketing and end-
2 user training; and Staff turnover.

3 The possible responses against each of the factors, in relation to their impact on
4 the maintainability of relational database-driven applications, were recorded on a
5 7-point bi-polar scale. The responses could be one of “high decrease”, “medium
6 decrease”, “low decrease”, “no impact”, “low increase”, “medium increase”, and
7 “high increase”. The scale was designed such that both the direction of the factors’
8 relationship (positive or negative i.e., whether a given factor increases maintain-
9 ability or decreases maintainability, respectively) and the strength of the relation-
10 ship with maintainability could be determined.

11 The third part of the questionnaire recorded opinions on which aspect of a rela-
12 tional database-driven application, from interface, data model, or algorithm objects/
13 computation intensive modules, was hardest to maintain; asked if any additional
14 predictors of maintainability; and recorded information on how a judgment is made
15 on the maintainability of relational database-driven applications in a company (for
16 which the option supplied was the amount of time to carry out a maintenance task).
17 Refer to Riaz [14] for the complete survey questionnaire.

18 The survey was pilot tested with four practitioners. Based on their feedback, the
19 survey was improved where the predictors from the SR, especially those that cor-
20 responded to various software metrics, were described using a simpler language in
21 order to be clearly understandable by the practitioners. Additionally, the second part
22 of the survey also indicated for the practitioners to assume the impact on main-
23 tainability due to an increase in the value (e.g., number of tables) or an improvement
24 in the quality (e.g., good software design or architecture), where applicable, of the
25 listed factors. Some practitioners also contacted us while filling out the survey form
26 to clarify the meanings of some of the factors and at a later stage regarding what to
27 assume in case of factors such as “good software design or architecture” in terms of
28 an increase in their value. We identified these as issues with the presentation of
29 information on the questionnaire and sent follow-up emails to all participants clar-
30 ifying the meanings of various factors and that an increase in the value in the context
31 of such factor relates to an improvement in the quality and that if in case of any
32 ambiguities they could contact us. Further, we also included the details of the factors
33 in the questionnaire. The issue of the assumption around the increase in the value of
34 a factor was identified at a very late stage and so we addressed this only by sending
35 follow-up emails. However, we did not receive any further emails from the partici-
36 pating practitioners and so it is safe to assume that this issue was corrected.

37 38 **3.2. Survey process**

39 In order to conduct the survey, approval from the University of Auckland Human
40 Participants Ethics Committee (UAHPEC) was obtained (Ref. 2009/527). The
41 process prescribed by UAHPEC was to contact the competent authorities within
42 the companies which would then extend the invitation to their employees. The
43

1 invitations were sent to more than 50 companies in New Zealand and more than 25
2 companies in Pakistan through email. The email addresses were obtained through
3 companies' listings on the Internet. In addition, a mailing list of the University of
4 Auckland's Center of Software Innovation (CSI) and authors' own contacts were also
5 used to make contact with the companies. The email invitation explained the pur-
6 pose and nature of the survey, provided the Web link to the survey, and asked the
7 companies' competent authorities to request participation by those employees within
8 their companies that had experience with developing and/or maintaining relational
9 database-driven software applications.

10 The survey and the relevant UAHPEC documents, which included a Participant
11 Information Sheet (PIS) and consent form both for the competent authority and
12 participating employee, were made available online. The PIS included details rele-
13 vant to the research. The other worth mentioning information that the PIS gave to
14 the participants include the conditions around the confidentiality and withdrawal
15 from participation. It stated that the confidentiality of the company or the partici-
16 pant will not be compromised and that the participants had the right to withdraw
17 their participation anytime within the three months of filling out the survey form.
18 The survey forms were filled out via a Web interface [14]. Some respondents also
19 downloaded the questionnaire and returned an e-copy. The survey was conducted
20 between June 2010 and September 2010.

21 22 **3.3. Data source**

23
24 The data source used herein comprised responses of a total of 40 software profes-
25 sionals — 13 from 8 different software companies in New Zealand and 27 from 15
26 different software companies in Pakistan, where one respondent was self-employed.
27 The roles of the participants varied from "Software Developer" to "Head of Devel-
28 opment". The participants had at least 2 years of experience with software devel-
29 opment and/or maintenance, and had an average experience of 7.8 years with a
30 minimum of 2 and maximum of 28 years of experience.

31 The two countries, Pakistan and New Zealand, were chosen for the following
32 reasons: (i) the interviews of which this research is informed were also conducted in
33 New Zealand and Pakistan; (ii) the results of the interviews established that the
34 context of this research was not culture-sensitive as the focus was on software
35 applications in general and not specific aspects related to people or their way of
36 carrying out their work; and (iii) the first and third authors are originally from
37 Pakistan and research was carried out at the University of Auckland, New Zealand
38 to which the first three authors are affiliated.

39 In terms of the participating companies, only 3 companies, from which only one
40 respondent each participated, were created 1, 2 and 5 years ago. All the other
41 companies were at least 10 years old. In addition, all companies except two were
42 engaged in development or maintenance of applications where 90% or more of their
43 applications had relational database backend. The other two also were involved in

developing or maintaining relational database-driven applications; however, most of their work did not involve this type of development. In relation to the use of DBMS, all the participating companies used either Oracle Database or Microsoft SQL Server.

3.4. *Data analysis*

Prior to analyzing the data, it was consolidated from two different sources, a database residing on Microsoft SQL Server used for the online survey and e-copies of the survey forms filled by some respondents for which the data was manually entered. Frequency analysis was the technique used for data analysis. It was well suited to the purpose as the intention was to see how many respondents thought of a predictor as having an impact 1 to 7 (high decrease to high increase) on the maintainability of relational database-driven software application.

During the frequency analysis, initially only the predictors that were reported to have an impact of “high decrease” or “high increase” on the maintainability of relational database-driven applications were considered. However, the frequency values did not stand out when only the extreme ends of the bi-polar scale were considered. For instance, for most of the factors the proportion of the respondents (out of 40) that chose one of the extreme ends of the scale (1 and 2 OR 6 and 7) were half or less than half i.e., between 15 and 20 (out of 40 respondents) and the proportion of the respondents that chose other options “medium decrease” to “medium increase” was equal or higher to those that selected one of the values at the extreme ends of the scale. Therefore, for final analysis, the sum of the frequencies for “high decrease” and “medium decrease” (1 and 2 on the bi-polar scale), and “medium increase” and “high increase” (6 and 7 on the bi-polar scale) were considered. The sum of the frequencies for the two ends of the bi-polar scale presented a better choice and represented better insights into the data.

4. Results

4.1. *Maintainability predictors*

The results of the survey data analysis corresponding to the second part of the survey are presented in Fig. 1, and show only those maintainability predictors for relational database-driven applications that had frequency values of 21 or higher i.e., those predictors that were reported by more than 50% of the respondents as having a high impact on the maintainability of the mentioned applications. Note that, the predictors presented in the table are all those predictors that had an impact of “medium increase” or “high increase” on the maintainability of relational database-driven applications. None of the predictors that had an impact of “medium decrease” or “high decrease” on maintainability were selected as a result of the frequency analysis. This shows that all the predictors listed in Fig. 1 have a positive relationship with the

10 *M. Riaz et al.*

Fig. 1. Predictors of maintainability for relational database-driven software applications.

maintainability of relational database-driven software applications i.e., they all result in an increase in the maintainability of relational database-driven applications whenever they also increase.

The number of selected predictors for each category (as defined in the survey questionnaire and mentioned in Sec. 3.1) against the total in that category are given in Table 1. Note that predictors belonging to categories “Size, Scope, and Complexity of Relational Database-Driven Software Applications” and “Web Applications with Relational Database Backend” were not selected during frequency analysis and are therefore not presented in the results discussed further.

It is also worth noting that the best predictor relates to the relational database schema. In addition, other predictors, such as good database design, optimal use of DBMS features, are also among the top predictors of maintainability for relational database-driven software applications. Note that these predictors were not reported in the studies selected in the SR yet these rank higher than the predictors that were identified both by SR and interviews.

The percentages of predictors selected for each category presented in Table 1.

Table 1. Percentage of maintainability predictors per category.

Category	Frequency	Total in category	Selection percentage
Database Schema	4	19	21%
Relational Database-Driven Applications	7	15	47%
Design Aspects of Database-Driven Applications	4	9	44%
Quality, QA, and Testing	17	20	85%
Change and Maintenance	4	10	40%
Development and Deployment Environment	4	7	57%
Organizational Culture and Policies	2	4	50%
Team, Team Communication, and Project Management	2	7	29%

The results given in Table 1 show that the most important category of predictors was “Quality, Quality Assurance, and Testing”, both in terms of the total number of predictors selected for overall analysis and number of predictors selected from the total number of predictors within the category. The second important category in terms of the number of factors contributing to maintainability prediction is “Relational Database-Driven Applications”. Other important categories relate to “Development and Deployment Environment” and “Organizational Culture and Policies”. The results clearly indicate that the factors related to quality, database schema, relational database-driven application play an important role in predicting maintainability. However, the results also indicate that the factors related to development environment and organizational culture are also considered important by practitioners in order to determine the maintainability of relational database-driven applications.

4.2. *Ranking of the aspect of relational database-driven applications for ease of maintenance*

The first section of the third part of the survey asked the respondents to assign a ranking of 1 (easiest) to 3 (hardest) to each of three listed aspects of the relational database-driven applications i.e., application’s interface, application’s data model, and application’s algorithmic objects or computation intensive modules. The results (see Fig. 2) show that the application’s interface was considered the easiest to maintain by 33 respondents, followed by the application’s data model (24 respondents) and algorithmic objects (23 respondents). As such, there was not too much difference between the number of respondents for second and third rankings, which suggest that the data model of a relational database-driven software application can be as difficult to maintain as algorithmic or computation intensive objects.

In addition to the ranking against the application’s aspects, this part of the survey also asked if the respondents wanted to report any other factors they believed were important for predicting maintainability of relational database-driven applications. Only three respondents added their responses, which included software development process level used in an organization; non-technical aspects related to customer

12 *M. Riaz et al.*

Fig. 2. Ranking of the aspects of relational database-driven applications as per ease of maintenance.

behavior which drives the timelines for performing maintenance tasks; and conformance of DBMS to standards, reliability of WANs, and impact of other software installed on the system such as anti-virus, operating system, etc. As such, these factors were not considered significant as part of the analysis as each of these were reported by only one respondent and would have ultimately been eliminated from the list of top maintainability predictors.

4.3. Measures of maintainability for relational database-driven software applications

The second section of the third part of the survey was aimed at recording the most used measure of maintainability among time to carry out a maintenance task, own experience and judgment, and an equation or a more sophisticated method. 13 respondents provided no answer to this question. Among the 27 responses received, 20 favored expert judgment, 4 favored time to carry out a maintenance task, 1 favored a more sophisticated method, and two reported a combination of time to carry out a maintenance task and their judgment. The only respondent who mentioned an equation or more sophisticated method also added a note to indicate that there is no use for an equation or model but the company uses mature practices so it has its own parameters to assess maintainability.

Overall, their own experience and judgment was the measure reported by most surveyed practitioners to measure the maintainability of relational database-driven applications. This suggests that maintainability of relational database-driven applications is determined by subjective assessment in practice.

5. Discussion

This paper presents the results of a survey conducted with software professionals. The survey was informed by a previously conducted SR [12] and the results of

1 interviews conducted with software professionals [13]. It was aimed at identifying
2 maintainability predictors that had the highest impact on the maintainability of
3 relational database-driven software applications. A list of 120 predictors belonging to
4 10 different categories was used on which expert opinions from software professionals
5 were recorded. The survey results suggest that the top 5 predictors related to
6 “Database Design”, “Design Aspects of Relational Database-Driven Applications”,
7 “Team, Team Communication and Project Management”, and “Change and
8 Maintenance”. The number of predictors selected by more than 50% respondents was
9 highest for the category “Quality, Quality Assurance, and Testing” followed by the
10 predictors related to “Relational Database-Driven Application”.

11 The main contribution of this paper is that it identifies the top predictors of
12 maintainability for relational database-driven applications. Other contributions in-
13 clude establishing and supporting by evidence that the type of applications most
14 developed in software organizations have a relational database backend, as discussed
15 in Sec. 3.3; interface of a relational database-driven application is considered easiest
16 to change by software practitioners whereas data model and computation intensive
17 modules are considered equally difficult to maintain in comparison to interface; and
18 maintainability in practice is measured using expert judgment.

19 The results of the survey presented some interesting findings in terms of the factors
20 considered important by software practitioners. From the results of the SR, the most
21 frequently mentioned predictors related to coupling, complexity and size related
22 measures. However, the survey results suggested that the factors related to “Size,
23 Scope, and Complexity of Relational Database-Driven Applications” were not
24 ranked higher than predictors belonging to other categories by the surveyed software
25 practitioners. In fact, the list of predictors ranked higher by more than 50% of the
26 respondents does not include any factor from this category. This is a very important
27 finding suggesting a huge gap between research and practice with regards to the
28 perception of maintainability predictors in the context of relational database-driven
29 applications. The possible reasons for this can be that most research experiments
30 reported in literature are not performed in an industrial setting and the applications
31 used in research experiments are generally small in scope. This makes it easier to
32 gather metrics and experiment on these applications whereas in practice there is lesser
33 room for experimentation due to tight deadlines. In addition, in practice no sophis-
34 ticated techniques for maintainability prediction are used and therefore, practitioners
35 rely more on the visible factors on which that they can more easily formulate a
36 judgment on maintainability. Also, the sample size for this research is not large so
37 there is also the possibility of lack of external validity. We believe that irrespective of
38 the reasons, there is a need to further investigate maintainability predictors of for
39 relational database-driven applications. In this regard, our work in progress involves
40 gathering project specific data on the predictors identified herein and creating
41 maintainability prediction models based on the gathered data. This, in addition to
42 providing prediction models, will also be a means to validating the predictors and will
43 suggest a need to consider predictors other than the ones selected herein.

1 The maintainability predictors for software applications suggested in the litera-
2 ture involved measures related to coupling, complexity and size of the application.
3 The results of the survey suggest that size related measures are not important pre-
4 dictors whereas predictors related to database design are very important for main-
5 tainability prediction of relational database-driven applications. This suggests that
6 the maintainability of relational database-driven applications is different from that
7 of the software applications.

8 It may appear that the data analysis method used is not very sophisticated and
9 rigorous. However, the purpose of this survey was to identify top predictors from the
10 list of 120 predictors such that these could be used for future work involved in this
11 research. The analysis technique used perfectly suited the purpose and the use for a
12 more sophisticated technique was hence, not required. Moreover, there are examples
13 in the literature on software prediction where frequency analysis has been used to
14 rank the predictors to be used further for the creation of prediction models [10].

15 It should also be noted that all the factors listed in the survey questionnaire were
16 sourced either from the results of the SR or the interviews conducted as previous
17 phases of this research. While some factors may not seem to be ideal candidates as
18 predictors of maintainability, e.g., effort and cost spent on maintenance, these were
19 included to avoid researchers' selection bias based on their own judgement. Addi-
20 tionally, because the maintainability is predicted in practice using expert judgment,
21 such predictors may hold value especially when maintainability is assessed based on
22 similar projects carried out in the past.

23 The possible limitations of this research are related to an unequal number of
24 respondents from the two countries and smaller sample size. While an equal number
25 of respondents from both countries would have provided better opportunity for data
26 analysis and comparison of results, the research itself was context free as established
27 by the results from the interviews [13]. Therefore, we believe that the unequal
28 number of respondents in this case is not a threat to the validity of the research.
29 With regard to small sample size, we believe that it poses threats to the external
30 validity of the research as the chances of obtaining different results from a larger
31 sample size cannot be ignored. However, considering the number of companies to
32 which invitations to participate were emailed, the length of the survey, the time
33 frame within which the survey had to be completed, and the fact that 34 profes-
34 sionals tried to fill out the survey questionnaire but dropped out before completing
35 it; we believe that the sample data is of a considerable size to derive valuable con-
36 clusions. We are also aware that the data gathered did not take into consideration
37 the characterization of the companies in terms of the company size, multi-national,
38 or Government agencies, etc. While we believe that this characterization may have
39 an impact on the overall processes followed within the companies, it is not of huge
40 significance in relation to the maintainability of the relational database-driven
41 applications. Nonetheless, it may be an interesting area to investigate whether or not
42 different types of companies perceive the maintainability and maintainability pre-
43 dictors differently.

6. Conclusions

This paper presents the results of a survey conducted with software professionals. The survey was informed by the results of a previously conducted SR and interviews conducted with software professionals. The aim of the survey was to rank the 120 maintainability predictors of relational database-driven applications belonging to 10 different categories for their strength in predicting maintainability. Out of 120 total factors, 44 factors were found to have an impact of either “high increase” or “medium increase” on maintainability, as per more than 50% of the respondents. The most important category of predictors was “Quality, Quality Assurance and Testing” whereas the top predictor “Correct definition of entities” belonged to the category “Database Design”. Our results also suggest that the practitioners measure maintainability using expert judgment and consider computation intensive modules of relational database-driven applications as easy or difficult to maintain as the data models.

Future work involves creating maintainability prediction models for relational database-driven software applications. In this regard, the work on gathering project related data on the predictors is already in progress.

References

1. P. Bhatt *et al.*, Influencing factors in outsourced software maintenance, *ACM SIGSOFT Soft. Eng. Notes* **31**(3) (2006) 1–6.
2. A. Cleve, T. Mens and J. Hainaut, Data-intensive system evolution, *Computer* **43**(8) (2010) 110–112.
3. E. H. Ferneley, Design metrics as an aid to software maintenance: An empirical study, *J. Softw. Maint. Res. Pract.* **11** (1999) 55–72.
4. S. K. Gardiokiotis, A structural approach towards maintenance of database applications, *IDEAS '04* (2004), pp. 277–282.
5. M. Genero, J. Olivas, M. Piattini and F. Romero, Using metrics to predict OO information systems maintainability, *ICAISE*, 2001, pp. 388–401.
6. IEEE Std. 610.12-1990, *Standard Glossary of Software Engineering Terminology* (IEEE Computer Society Press, Los Alamitos, CA, 1993).
7. G. M. Kapfhammer and M. L. Soffa, A family of test adequacy criteria for DB-driven applications, *ESEC/FSE 2003* (2003), pp. 98–107.
8. J. S. Lim, S. R. Jeong and S. R. Schach, An empirical investigation of the impact of the object-oriented paradigm on the maintainability of real-world mission-critical software, *J. Syst. Software* **77** (2005) 131–138.
9. A. Maule, W. Emmerich and D. S. Rosenblum, Impact analysis of database schema changes, *ICSE '08* (2008), pp. 451–460.
10. E. Mendes, N. Mosley and S. Counsell, Investigating web size metrics for early web cost estimation, *J. Syst. Softw.* **77**(2) (2005) 157–172.
11. P. Oman and J. Hagemester, Construction and testing of polynomials predicting software maintainability, *J Syst Software* **24** (1994).
12. M. Riaz, E. Mendes and E. Tempero, A systematic review of software maintainability prediction and metrics, *ESEM 2009* (2009), pp. 367–377.

AQ: Please provide the page range.

16 *M. Riaz et al.*

- 1 13. M. Riaz, E. Mendes and E. Tempero, Towards maintainability prediction for relational
2 database-driven software applications — evidence from software practitioners, *ASEA*
3 *2010* (2010), pp. 110–119.
- 4 14. M. Riaz, Survey of software maintainability prediction for relational database-driven
5 software applications, http://www.cs.auckland.ac.nz/survey_2010.
- 6 15. M. Seltzer, Beyond relational databases, *Communications of ACM* **51**(7) (2008) 52–58.
- 7 16. Y. Deng and D. Chays, Testing database transactions with AGENDA, *ICSE 2005* (2005),
8 pp. 78–87.
- 9 17. E. F. Codd, A relational model of data for large shared data banks, *Commun. ACM* **26**(1)
10 (1983) 64–69.
- 11 18. T. J. Heintz, Object-oriented databases and their impact on future business database
12 applications, *Information and Management* **20**(2) (1991) 95–103.
- 13 19. M. Riaz, E. Mendes and E. D. Tempero, Maintainability prediction for database-driven
14 software applications — preliminary results from interviews with software professionals,
15 *SEDE 2010* (2010).
- 16 20. R. R. Barton, Implementing a relational database for an accelerated-life-test facility,
17 *IEEE Trans. Reliability* **43**(1) (1994) 11–21.
- 18 21. M. Riaz, E. Mendes and E. D. Tempero, Maintainability predictors for relational
19 database-driven software applications — results from a survey, *SEKE 2011* (2011).
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35
- 36
- 37
- 38
- 39
- 40
- 41
- 42
- 43

AQ: Please
provide the
page range
for Refs. 19 &
21.