

Fuzzy-Provenance Architecture for Effort Metric Data Quality Assessment

Rita Cristina Galarraga Berardi, Duncan Dubugras Alcoba Ruiz

Faculdade de Informática – Pontifícia Universidade Católica do Rio Grande do Sul
(PUCRS)

Caixa Postal s/n - 90.619-900 – Porto Alegre – RS – Brazil

{rita.berardi,duncan}@pucrs.br

***Abstract.** Software companies rely on stored metric data in order to track and manage their projects, through analyzing, monitoring and estimating software metrics. If managers cannot believe the metrics data, the product that is being developed is fated to fail. Currently, the assessment of software effort is subjective and derived mainly through managers' assumptions, which is fundamentally an error-prone process. We present an architecture for assessing data quality of software effort metric based on data provenance associated with a mechanism of logical inference (fuzzy logic). The contribution is to provide an assessment to search evident reasons for a low quality in order to ensure that the metrics can be used with sufficient reliability.*

1. Introduction

Lack of data quality is a current issue within enterprises due to their strategy of focusing on storing great amounts of data without any proper quality control. Low metric data believability and problems of predicting costs is the result of such approach, since decision making tasks rely on stored data and the quality they possess.

Considering software companies, they store large amounts of data as a result of the software development process measurement task, given that maturity models such as CMMI (Capability and Maturity Model Integration) argue in favour of collecting metrics in order to monitor, analyze and continuously improve the development process [SEI 2006].

Managers of software companies, however, should not doubt the believability of the metrics that are collected specifically to aid their decision-making process. This is the case of the effort metric, which measures the time spent by a software team to develop, maintain and fix defects of a given product [IEEE 1998]. It is also of the team's interest that managers believe the veracity of metrics data, once such data can be used for tasks assignment within the team. If a manager underestimates the effort spent by her team considering previously stored data, then the software team may be under considerable pressure to finish the product quickly, and hence the resulting software may not be fully functional and well tested.

Currently, the assessment of software effort data quality is totally based on the manager's assumptions and sustained by the relationship of trust between the manager and her team. This kind of relationship is essentially flawed because it is based on

human actions. Although the manager experience on analyzing the data is frequently efficient, this assessment strategy does not present clear evidences that could identify where potential problems are allowing certain aspects that have strong influence on quality to be fixed or improved.

Software effort data are especially complex to be evaluated since there are no mechanisms to either automate their capture or identify eventual measurement flaws. Effort data are recorded by team members (usually measured in hours), and since it is a manual process, it is subject to problems intrinsic to the human nature.

Another conflicting factor that raises the complexity on evaluating the quality of software effort data is a well-known issue of data quality: the real definition of *high* or *low* quality. Such data are within different contexts and formats, and under distinct measurement goals, which adds significant complexity on defining whether the captured values are of high or low quality.

Considering all the aforementioned problems, in this paper we present a architecture for evaluating software effort data, which takes into account different circumstances and roles that take part of the software development process. Nonetheless it should be noted that we do not present a architecture for dealing with effort metrics, but for managing effort metric data. Check the IEEE 1061 standard [IEEE 1998] for defining and dealing with software metrics. The main contribution is to provide an assessment of data quality metrics of effort in Software Development Process, seeking evident the reasons for a low quality. Having a model of inference it is possible to assign levels of quality to the data, and thus to enable the identification of those that are actually useful to a reliable decision-making. This work is organized as follows. First, we present the requirements we believe an architecture for evaluating software effort data should fulfil. Then, we present the related approaches we have found in the literature. Afterwards, we present our architecture and how its components are built. And finally, we present our conclusions through a objective evaluation of what was done, appointing the research strengths, weaknesses and opportunities for improvement.

2. Architecture Requirements

Based on the previously pointed issues and the context where the data is inserted, it is possible to point out some essential aspects that should be part of an assessment architecture:

- **Granularity:** the architecture should be capable of providing data quality assessment within the recording level of effort data, because such level enables the identification of problems, which improvements that could be incorporated to the effort data;
- **Functionality:** the architecture should provide a reasonably practical approach for data analysis in order to allow fast answers to the final user;
- **Non-crispy assessment:** the architecture should avoid evaluating the data through a crispy approach, because we consider that within the software metrics context, it is quite difficult to evaluate values in a true or false fashion, especially considering the several dimensions beneath such dataset, which is detailed later;

- Objectivity: the architecture should point out objectively the assessment result, as well as the set of criteria used to induct such result;
- Context: the architecture should take into account the software effort data context, in order to better consider the intrinsic characteristics of the software development process (e.g. life cycle, manual collection of metrics);

All these requirements should be considered of high importance for defining an architecture to evaluate software effort data and we may derive a flawed architecture with low applicability and utility by ignoring them.

3. Related Work

Lee et al. (2006) describe the techniques for evaluating data quality, and among them we can mention: (1) Data Quality Survey, (2) Data Quality Metrics and (3) Data Integrity Analysis. Considering such techniques, we have studied different researches that explore them and their main features.

For *Data Quality Survey*, the following works were investigated. Lee et al. (2002) define a methodology (namely, AIQM) that makes use of surveys to evaluate data quality at any context whatsoever. Such methodology has three main components: (1) dimensions definition for evaluation; (2) data collection through surveys; and (3) analysis techniques to identify data quality problems. Berry et al. (2004) suggest the use of two different kinds of assessment instruments, the first one being a generic survey and the second a survey focused on software measurement data. The goal in such evaluation is to improve the measurement process and not the data itself.

Researches on *Data Quality Metrics* are explored next. Pipino et al. (2002) present a set of metrics described in a methodology (DQA) based on the dimensions presented by [Wand and Wang 1996], formulated through three calculation forms (simple ratio, min/max operations and weighted average) which indicate the quality of a generic dataset. These authors also present the possibility of combining such metrics to a subjective evaluation in order to enhance data quality. Caballero et al. (2007) specify a measurement model (DQMIM) which considers the user needs to define *what should be measured, how and why*, as well as *who* should be doing it. Based on these definitions, a set of metrics to evaluate the quality of generic datasets was developed. Batini et al. (2007) report an assessment methodology (ORME-DQ) that consists of four phases: (1) environment analysis; (2) critical dataset selection; (3) quantitative and qualitative quality evaluation; and (4) continuous quality evaluation. During phase three, the metrics are used for quality evaluation, and their format depends on the respective dimension (simple string comparison, distance calculation, etc). Prat and Madnick (2008) define an approach for measuring believability of a dataset (MDB: PA). The authors created a set of metrics that are used over a database which tracks the *data provenance*. Both metrics and provenance model can be applied in general purpose.

Considering *Data Integrity Analysis*, Lee et al. (2004) present an interactive process for quality improvement (PEDI) through the adoption of integrity constraints, which perform over different dimensions: (1) Column integrity: accuracy and interpretability; (2) Entity integrity: accuracy and completeness; (3) User-defined integrity: consistency.

By analyzing these researches, we classified how each one covers the requirements we have identified to build an assessment architecture for evaluating software effort metric data quality. Table 1 shows this rationale, where the darkest colour indicates that the related work fully covers the requirement, whereas the lightest one indicates that it supports the requirement but not according to what is expected and defined in this paper, and the lack of colour indicates non-attainment of the requirement at all.

Table 1. Related work and conformity to the architecture building requirements

Related work Requirement	Data Quality Survey		Data Quality Metrics				Data Integrity Analysis
	AIQM (2002)	ASM-IQS (2004)	DQA (2002)	DQMIM (2007)	ORME-DQ (2007)	MDB:PA (2008)	PEDI (2004)
Granularity							
Functionality							
Non-Crispy							
Objectivity							
Context							

By inspecting Table 1, we have not found any research able to fulfill all requirements. We can notice, for instance, that the Data Quality Survey technique would not be applicable due to its mechanisms of application and results analysis. The time spent during the answers' collection and analysis phase would severely slow down the process. Despite good functionality, the Data Quality Metrics technique is not suited for effort data because of granularity issues, and also because metrics usually present a dichotomist evaluation (good or bad, high or low) considering a group of data (e.g., 70% of the X dataset has low quality). Data Integrity Analysis was discarded as a possible approach because it does not take into consideration effective context information, which is essential when analyzing the factors that could influence effort data quality.

Since the approaches suggested by Lee et. al. (2006) were not suitable for being used as basis of an evaluation architecture for effort data, other alternatives were searched for evaluating data quality. Liebchen et al. (2007), for instance, consider data quality evaluation of software metrics through data mining classification. However, since classification requires a data pre-processing stage, which may alter the real meaning of the dataset that is being analyzed, we have discarded such approach in order to avoid missing important data details. The research carried out by Caro (2007) considers the empirical knowledge of website users to identify criteria for evaluating websites. A Bayesian network is built over these subjective criteria, indicating the probability each dataset has on being of high/low quality. Finally, Data Provenance is another approach commonly used in the search of data quality, achieved through tracking the process that provides the data and the corresponding data sources. Different application domains were explored through Data Provenance, such as Astronomy, Biology and Physics, in the work of Foster et. al. (2002), and Agriculture in the work of Fileto et. al. (2003). Widom (2005) and Buneman (2001, 2006) also explore Data

Provenance in Biology data. As far as we are concerned, Data Provenance has not been used for software metrics quality evaluation.

4. Fuzzy-Provenance Architecture for Effort Metric Data Quality Assessment

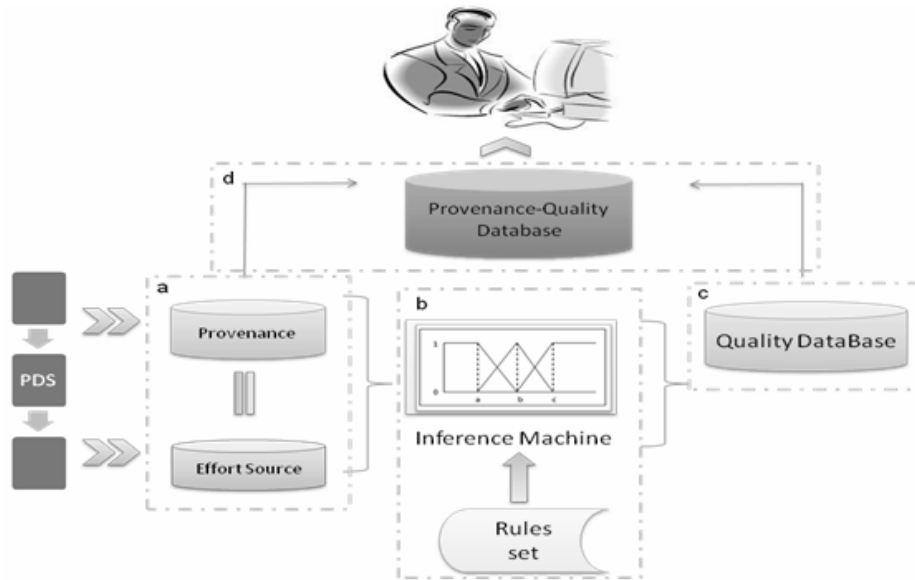


Figure 1. Fuzzy-Provenance Architecture

4.1. Architecture

The architecture we have developed for evaluating the quality of software effort data is presented in Fig.1. We have divided it into four main components (a, b, c and d).

- a. **Provenance Component:** responsible for storing metadata traceability in a Provenance Database. When effort becomes recorded, the Provenance Database tracks metadata which capture the record circumstances, either from the Effort Source Database or from the software process development environment. This trace and storage process happens whenever the Effort Source database is modified: insertion or update.
- b. **Inference Machine Component:** is represented by an inference machine that makes use of a previously created rule set. According to the frequency chosen by the enterprise (daily, weekly, monthly, etc.) the inference machine receives the provenance data that corresponds to the time interval desired and through the rules set it evaluates the quality of each effort record, considering all possible modifications an effort record may have been through.
- c. **Quality Database Component:** is represented by a Quality Database that stores the output of the inference machine. The Quality Database contains the evaluation of each effort register (identified by an ID) considering each quality dimension and presents the level of adherence of an effort record to the quality levels Low (L), Reasonable (R), Acceptable (A) and High (H).
- d. **Provenance-Quality Database Component:** is represented by a Data Warehouse that aims to provide analysis resources for the software company management.

The full history of effort records is stored in this Data Warehouse, allowing tendency analysis and several other functionalities;

Merging these four components in a single architecture enables the software company to analyze the current state of the effort data, as well as to identify flawed points and improvement margins. Each component has an important role within the architecture, but it is the combination of all components that synthesizes the contribution of this work.

4.2. Building the Architecture Components

Assessment of data quality requires assessments through a number of dimensions [Lee et. al 2006], see Table 2. Each company should determine which dimensions are important according to the context involved and also which variables that constitute each dimension should be defined.

Table 2. Architecture dimensions

Dimension	Description
Accuracy	The extent to which a data value is correct.
Consistency	The extent to which a data value is consistent along the data transformation cycle.
Completeness	The extent to which a data value is complete considering the company objectives.
Timeliness	The extent to which a data value reflects how up-to-date it is with respect to the task for which it is being used.

Our approach makes use of data provenance variables to constitute each dimension, because we believe that by tracking the effort data during the whole data cycle, we are able to infer levels of quality for these data. This approach is a novelty in this area since no similar work was found to the best of our knowledge.

It is necessary to identify which provenance variables must be used and how they relate to each dimension. We believe that such information is already known within the company, because even though the current assessment mechanism is subjective, the stakeholders make use of certain criteria and variables that suit these criteria. Thus, our role is to capture this empiric knowledge in order to make an automatic evaluation possible, enabling the data evaluation process to belong to the company and not to any individual that works for the company.

With the purpose of capturing this empiric knowledge, we have chosen as research instrument the semi-structured interview, because we believe a survey would not be able to capture important details our research demands. The semi-structured interview methodology makes the information extraction easier because the presence of the interviewer allows new questions to be raised as consequence of a given answer [MARCONI, 2003].

Even if the interview is not a totally structured procedure, it is well-organized in order to extract crucial information for building the architecture. The interview document is divided in three parts. The first part is used to contextualize the interviewee

in order to verify the degree of knowledge she has concerning effort data quality. The second part consists in the identification of the importance degree associated to each dimension and quality level, which will be further explained in this paper. Finally, the third part describes each dimension through the provenance data, which is further explained. There is also an authorization section with the aim of enabling the answers provided to be used for researching purposes only and to keep the anonymity of the interviewee.

According to the interview methodology we have adopted, a previous test should be done with the intention of detecting possible unexplored points and unnecessary material prior to the real interview process with the company's collaborators. It is important, however, that the fictional interviewees must have some degree of domain knowledge. We have selected ten of our fellow researchers from the research group who also work in the partner enterprise which funds this research to answer this test.

After this test, we have invited ten company stakeholders, divided in three different profiles: collector, custodian and consumer. By dividing the stakeholders in these profiles, we are able to better understand the problems present in effort data over different visions along the data cycle, which is an important factor in increasing data quality awareness in the organization [Lee et. al. 2006].

Four of the stakeholders are part of the consumer profile (managers), four are collectors and two are custodians. Even though the number of interviewees may seem small, we consider it to be a satisfactory sample for the kind of adopted methodology, because we are not seeking for statistical analysis but to better aggregate knowledge and extract information. With that in mind, we have selected these stakeholders rigorously and not randomly, in order to choose employees with a high degree of experience in software process development, a high degree of project and data commitment and with a high degree of responsibility. To capture the stakeholders' needs is a hard task, but it is a good manner to understand the domain that usually brings satisfactory results [Redman 2001].

4.3. Building a Data Provenance Model

During the interview process, we have provided to each stakeholder a list with possibly useful provenance information for tracking effort data. For building the Provenance Component of our architecture, we have considered the provenance information which were selected as most important by the stakeholders from our list, and also additional variables that were not part of our initial list. Also, we have built a new provenance model because the ones that already exist in the literature do not take into account provenance data that are specific to the effort record context. Figure 2 depicts this provenance model in UML notation. The provenance model, along with the already existent project's source databases constitutes the Provenance Component of our architecture.

The Provenance database works as a tracking mechanism every time an effort data is either inserted for the first time in the Source database or when Source data are being updated. The tracking granularity is at the recording level: (a) where the data to be tracked refers to the information of who is updating the data, (b) historic data (date, new value, update dates), (c) the gap between the work day and the registering day, (d) SDP

phase data (name, start and finish date). The output generated by the Provenance Component is used as input to the *Inference Machine Component*.

Currently, the provenance which is being tracked by our approach refers to the circumstances of the data collection. Other kinds of provenance like source and calculus are not a concern of our research, because we are focusing only in the effort data and its variances over time.

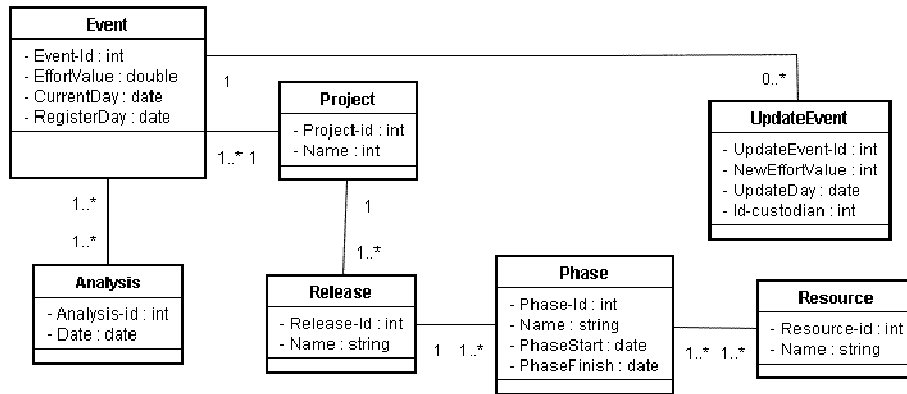


Figure 2. UML Provenance Model

4.4. Building the Inference Rules

The Inference Machine Component consists of a rules set and an inference engine that through membership functions processes these rules resulting in a degree of quality for the data. This machine has two processing levels. In the first one, it processes the adherence degree of the data considering each dimension, through the rules set and the membership functions, resulting in values between the data interval [0, 1]. The second level makes use of the values from the first level to calculate the data adherence degree for different quality levels, as illustrated in Figure 3.

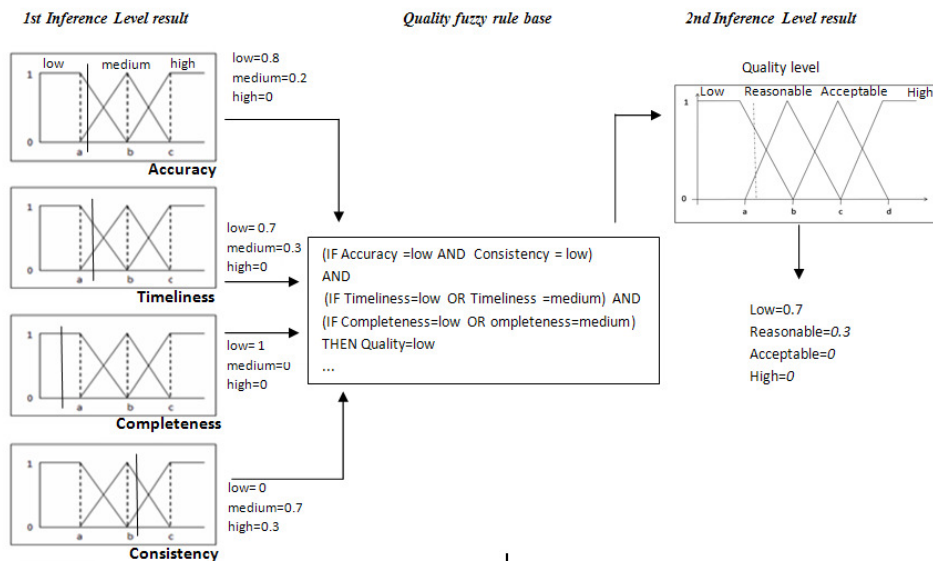


Figure 3. Inference Machine

The rules set and the respective membership functions were consolidated through the interview process. At first, each stakeholder had to explain the importance (low, medium or high) of each dimension for a given quality degree (low, reasonable, acceptable or high). For example, consider the following explanation given by one of the stakeholders: “if the data is of low accuracy and consistency, it will be consider as of low quality, and there is no need to check whether its completeness or timeliness are low or medium, because I consider accuracy and consistency more important than the other dimensions.” If such a statement was made, we could build the following rule:

IF data X has low accuracy AND has low consistency AND (has low timeliness OR has medium timeliness) AND (has low completeness OR has medium completeness) THEN X has low quality.

Following the interview process, the stakeholder checks which provenance variables from the list are related to each dimension, and then, through rules creation she relates these variables to constitute what characterizes a (low-medium-high) (accuracy-timeliness-completeness-consistency) data. For example, if the responsible for recording the effort data did not record her hours of work at the end of the work day, then probably the data has doubtful accuracy, and depending on how long she took to record her hours, this doubt considering the data accuracy tends to rise. In this particular case, the provenance variables are: work date (x), registering date (y) and difference between x and y (z). The rule block that can be built is:

*IF z is high THEN accuracy is low, IF z is medium THEN accuracy is medium
IF z is low THEN accuracy is high.*

Then, the stakeholder has to define, according to her experience and empiric knowledge, what should be considered as a high-medium-low value for z, and the intersection of levels, as can be shown in the membership function in Figure 4. The main advantage of processing rules through membership functions that result in values between [0,1] is the flexibility of the data classification [Altrock 1995].

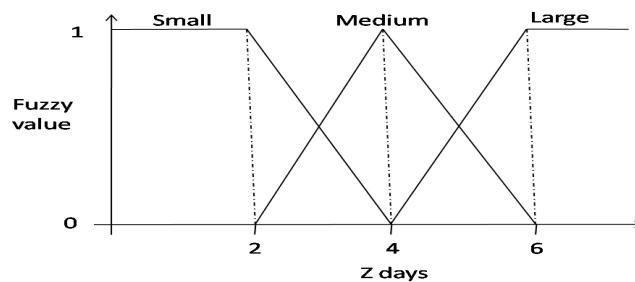


Figure 4. Graphic of a membership function.

The dimensions composed by provenance variables taken by the result of the interview are: (1) Accuracy: (#WD-RD), (#FDP-RD) and (#OV-UV); (2) Timeliness: (#WD-RD) and (#AD-CD); (3) Completeness: (#WD-RD) and (#FDP-RD); Consistency: (#OV-UV). Where (#WD-RD) is the difference between work day and the registering day, (#FDP-RD) is the difference between the finish date of the phase and the registering day, (#OV-UV) is the difference between the original effort value and the updated effort value and (#AD-CD) is the difference between the analysis date and the collection date.

The fuzzy values indicate how adherent is the data to each level, and not only a probability the data holds of being categorized in a given level. For example, the

information that the data is of low quality, but that it is almost in the margin of being considered reasonable, is much richer and useful than simply saying the data is of low quality. Besides, the fuzzy evaluation seems to suit better the metrics data context, whose complexity and dynamism are features that cannot be ignored during an evaluation process, making a crispy evaluation error-prone. It should be noted that, in the best of our knowledge, there is no research that makes use of fuzzy logic for data quality evaluation in the context of software metrics, what is another novelty of our research. However, fuzzy logic has been widely used in software estimation and software quality management researches.

4.4. Generating a Quality Database

With the results of the inference machine calculations that make use of data provenance, it is generated a database to store each effort value evaluation in each dimension to each quality level identified by ID, as can be seen in Figure 5. This is a simple component that is used as source of the analysis component *Provenance-Quality Database*. Below, we have an example of what could be a database tuple of this component.

ID	Accuracy	Timeliness	Completeness	Consistency	Low	Reasonable	Acceptable	High
01	0.2	0.1	0.6	0.8	0.1	0.6	0.8	0.4

Identification
Dimensions
Quality Level

Fuzzy Value

Figure 5. Generating a quality database.

4.5. Generating a Provenance-Quality Database

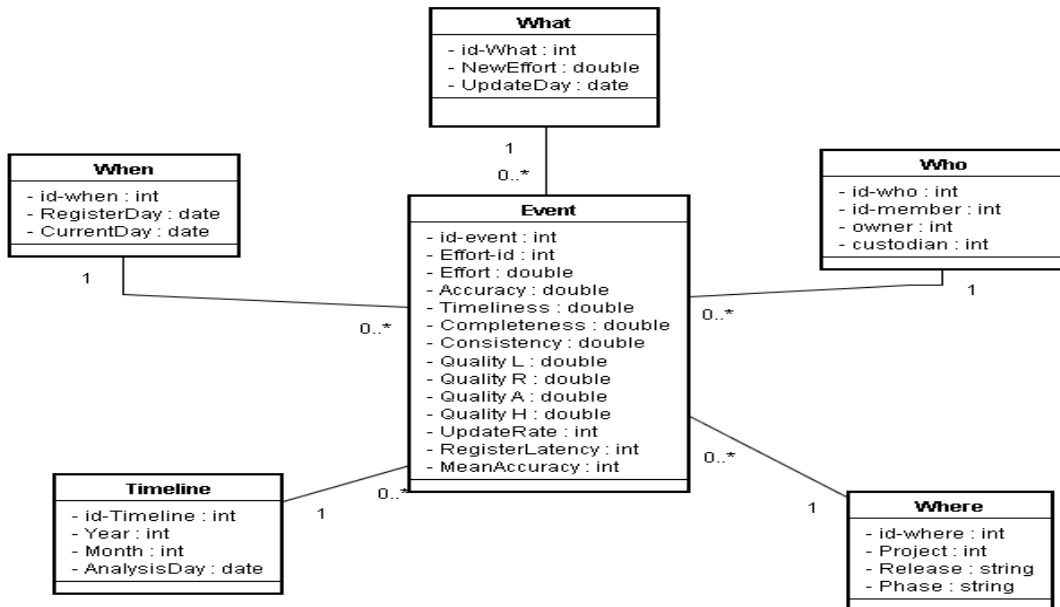


Figure 6. UML Quality Data Warehouse model.

The component that keeps the company evaluation’s historic and that enables tendency

analysis in order to indicate possible improvement actions is structured in the form of a data warehouse (Fig. 6). The Provenance-Quality Database is composed by a fact table (Event) that stores the values of the analyzed dimensions, and also the final consolidated quality level values. This fact table is surrounded by the dimensions *When*, *What*, *Timeline*, *Who* and *Where*. The Provenance-Quality component uses as sources the Provenance Database, Effort Source database and Quality Database, and it is the interface between managers and architecture. It is populated according to a company specified periodicity (weekly, quarterly or monthly). The only constraint for defining this periodicity is that the effort data must have already finished their cycle. That is, the evaluation process only happens for datasets that can no longer be modified, because our architecture does not support partial data evaluation.

5. Case study

The case study was conducted within the largest software maintenance project of a large software operation in Brazil. The project's name and all classified information were changed or suppressed in the following description. Project P1 is a continuous maintenance project that inserts, modifies and removes functionalities in a software product previously developed by the company. The software product is divided into versions, where each version corresponds to a single iteration in the maintenance lifecycle. An iteration must deliver a number of labor-hours/month to the client and may have one or more product releases. Each Service Order (*SO*) aggregates a set of change requirements for the product. The Project P1's teams are divided into: (1) Design Team, (2) Client Developers, (3) Server Developers and (5) Testers. It is expected that in the end of each work day each member of these teams record her hours spent in that day. Monthly, a member of one of these teams is allocated for collecting the data effort set and preparing a presentation to be analyzed by the P1's manager.

5.1. Experimentation Plan

An important task in the decision making process is to define how long the project will take. In this sense, data mining algorithms are executed for predicting this period. However, one of the problems faced in this execution is concerning data quality of the training dataset, which can strongly affect the prediction model generated by the algorithm. So, we considered this issue in our experimentation by defining the following hypothesis: "*All the evaluation measures of the prediction model are improved when the dataset effort data is submitted to the fuzzy-provenance architecture of assessment*". The evaluation measures we have mentioned are: correlation coefficient, root mean-squared error, mean absolute error, root relative squared error, relative absolute error, mean magnitude of relative error and prediction at level l .

For the verification of the hypothesis acceptance, we divided the experiment into three main steps:

- (1) Build a predictive model without having submitted the effort dataset to the architecture of assessment (the *original predictive model*).
- (2) Build a predictive model having submitted the effort dataset to the architecture of assessment (the *evaluated predictive model*).
 - (2.a) Submit the effort dataset to the architecture;
 - (2.b) Prepare the evaluated dataset to serve as input for the data mining algorithm, replacing the data evaluated as lower than Reasonable (i.e., Low) by

the average of the data evaluated as Reasonable or better (i.e., Acceptable and High).

- (3) Compare the evaluation measures on each predictive model.

After the comparison which is done in step 3, it is possible to infer if the hypothesis can be accepted or rejected.

5.2. Experiment's Execution

We have chosen the data mining model tree algorithm *M5P*, because of the following reasons: (a) Ease of use and understanding, (b) Suggested by renowned authors and (c) Availability of tools. The effort dataset consists of 1042 records grouped in 40 *SO* instances, each instance belongs to a unique *SO* of project P1.

In this experiment, we predict the effort spent so as to fulfill a *SO* cycle. Our main goal is to improve the evaluation measures of the predictive model generated by the data mining algorithm. As input for *M5P* we have: size of a *SO* (KLOC), adjusted function-points for each *SO*, estimated-Project (estimate for the project phase given by an expert), estimated-Server, estimated-Client, estimated-Integration, actual-Project (the actual effort spent for that *SO* in the project phase), actual-Server, actual-Client, actual-Integration, number of requirement documents for each *SO* (NDOCs), total-estimated, total-actual. The class attribute is total-estimated. We have considered all the attributes that refer to the estimates given by an expert because we believe that the data mining algorithm can improve the estimation task of the expert but it does not eliminate her from the process.

Having built the predictive models, the values of the evaluation measures of each one are exposed on the table 3.

Table 3. Evaluating measures of both predictive models

	Predictive Model 1	Predictive Model 2
Correlation Coefficient	0,923	0,9259
Mean absolute error	37,508	37,0781
Root mean-squared error	57,0924	55,1948
Relative absolute error	45,9478	45,0035
Root relative squared error	40,165	38,6572
Mean magnitude of relative error	0,6254	0,617
Prediction of level 25	0,375	0,55

Analyzing the results on the table 3 we can accept the hypothesis because all of the evaluation measures have shown improvement when the dataset was submitted to the architecture to be assessed before going through the data mining algorithm. We are aware that all the error values are very high despite of being better in the evaluated predictive model, but we still believe that the hypothesis can be accepted and more than that, we believe that the architecture has contributed with the improvement of the estimation task, which is a non-trivial task due to the doubts on the quality of the dataset. In addition we conducted another test statistic which is rather recommended by the renowned authors in the area of data mining: MMRE (Mean Magnitude of Relative Error) and PRED (25)[Witten and Frank 2005]. The results are shown on the table 4:

Table 4. Test statistic of both predictive models

	Predictive Model 1	Predictive Model 2
MMRE	0,625439	0,617775
PRED (25)	0,375	0,55

The results in table 4 show that although the error rate is high in both predictive models (predictive model 1: 62%, predictive model 2: 61%), the mistakes of the predictive model 2 are more within the range of error while the errors of the predictive model 1 are only 37 within the margin of error allowed. This analysis confirms the acceptance of the hypothesis.

Table 5: SWO analysis of the Fuzzy-Provenance Architecture

Strength	Weakness	Opportunities
<ul style="list-style-type: none"> - There are no similar proposals in the DQ literature and in the quality of software effort metrics; - The low granularity, which is not usually explored in related works, that tend to consider a dataset granularity level, instead of the data level itself; - Functionality suits the requirements specified. The assessment process is not time-consuming, which fits perfectly the managers' needs; - Non-crispy approach, allowing a more suitable data evaluation considering the effort data context; - Evaluation process allows visualizing how adherent certain data is to a given level, and how close it is from the other levels; - Provenance model suits the specificity of the effort data; - Modeling process guided according to what was described by employees that work every day with metrics. 	<ul style="list-style-type: none"> - Maybe gathering more stakeholders, from different companies, could result in more detailed information; - The need of adapting a fuzzy-logic tool to work within the architecture environment. 	<ul style="list-style-type: none"> - Improvement of the inference machine by adding a higher number of inference rules and membership functions; - Using other mechanisms to determine quality levels (Bayesian networks, for example); - Building a graphic interface to dictate the architecture behavior; - Refine the architecture through other real case studies. - Doing partial data quality evaluations, from the moment a given data value is recorded in the source database for the first time, allowing the analysis of how the data evolves over time.

6. Conclusion

The contribution of this research is aggregating evidences in the assessment of the effort metric through the use of data provenance, allowing a more consistent evaluation and a more objective improvement process, once it is identified where the lack of quality is. The use of well-defined dimensions makes the process of decision making for quality improvement easier because the identification of errors in specific points reduces the investigation and action-taking time spent to avoid such errors from keep happening.

Moreover, according to the quality level presented by the architecture, the data can be directed to different kinds of project tracking (analysis and prediction), that may present distinct quality-demanding levels. Other potential benefits are the reduction of

assessment time, considering that the architecture automates the process and enables richer analysis services. The different profiles that were taken into account for building the evaluation architecture have also enabled a domain understanding, and consequently the possibility of exploring a wider range of aspects in the software effort data quality. In addition, through the experiments executed in the case study it was possible to verify one of the real benefits of a proper evaluation process.

Finally, we have adopted the SWO analysis to point out the main features of our research (Table 5). Future directions of this work are based on the weakness and opportunities presented in the second and third columns of Table 5: first we intend to refine the membership functions and the rules set in order to achieve more faithful results. As future work, we plan to develop solutions to those opportunities, especially by unifying all architecture components into a single and powerful tool, because as far as it goes we have only tested solutions with disconnected independent components, which raises the throughput due to the need of navigating through different environments and tools. Also, we intend to integrate this architecture to another environment developed by the research group called SPDW [Becker et. al. 2006].

7. Acknowledgments

This work was possible due to the collaboration of HP EAS Brazil - Porto Alegre. This project was funded by the HP-PUCRS cooperation agreement.

8. References

- Altrock, C. V. (1995) *Fuzzy Logic & NeuroFuzzy Applications Explained*, Prentice Hall, p.342
- Batini, C., Barone, D., Mastrella, M., Maurino, A., Ruffini, C., (2007) A Framework and a Methodology for data quality assessment and monitoring. Proceedings of the Twelfth International Conference on Information Quality (ICIQ-07) MIT.
- Becker K., Ruiz, D., Cunha, V., Novello, T., and Souza, F., (2006) Spdw: A software development process performance data warehousing environment. In SEW '06: Proceedings of the 30th Annual IEEE/NASA Software Engineering Workshop, pages 107-118, Washington, DC, USA. IEEE Computer Society.
- Berry, M., Jeffery, R., Aurum (2004) Assessment of Software Measurement: an Information Quality Study. Proceedings of the 10th International Symposium on Software Metrics. Pp 314- 325
- Buneman, P., Chapman, A., Cheney, J. (2006) Provenance management in curated databases. In SIGMOD '06: Proceedings of the 2006 ACM SIGMOD International Conference on Management of data, pages 539-550, New York, NY, USA
- Buneman, P., Khanna, S., Tan, Wang-Chiew. (2001) Why and where: A characterization of data provenance. Lecture Notes in Computer Science.
- Caballero, I. and Verbo, E. (2007) A Data Quality Measurement Information Model Based on ISO/IEC 15939. Proceedings of the Twelfth International Conference on Information Quality (ICIQ-07) MIT, 2007.
- Caro, A. (2007) A Probabilistic Approach to Web Portal's Data Quality Evaluation. Proceedings of the Sixth International Conference Information on the Quality of

Information and Communications Technology (QUATIC 2007), publisher IEEE Computer Society

- Fileto, R., Medeiros, C.B., Liu, L., Pu, C., Assad, E. D. (2003) Using domain ontologies to help track data provenance. SBBD, pp. 84_98
- Foster, I., Voekler, J., Wilde, M., Zhao, Y. (2002) Chimera: A virtual data system for representing, querying, and automating data derivation. Proceedings of the 14th International Conference on Scientific and Statistical Database Management. Pages: 37 - 46.
- IEEE Computer Society (1998) IEEE Standard for Software Quality Metrics Methodology. The Institute of Electrical and Electronics Engineers, Inc., New York, December.
- Lee, Y. W., Pipino, Leo L., Funk, James D., Wang, Richard Y. (2006) *Journey to Data Quality*, The MIT Press, Cambridge, Massachusetts, London, p.226
- Lee, Y. W., Pipino, L., Strong, D. M., Wang, Richard Y. (2004) Process-embedded data integrity, *Journal of Database Management*, 15(1), pp 87-103.
- Lee, Y. W., Strong, D. M., Kahn, B. K., and R. Y. Wang. (2002) AIMQ: A Methodology for Information Quality Assessment. *Information and Management* 40 (2): 133-146.
- Liebchen, G., Twala, B., Shepperd, M., (2007). Filtering, Robust Filtering, Polishing: Techniques for Addressing Quality in Software Data Empirical Software Engineering and Measurement, ESEM. First International Symposium on Empirical Software Engineering and Measurement.
- Marconi, M., Lakatos, E., 2003 Fundamentos de Metodologia Científica. Editora Atlas S.A. 5 edição.
- Pipino, L. L, Lee, Y. W., Wang, R. Y. (2002) Data Quality Assessment, *Communications of the ACM*, 45 (4), pp211
- Prat, N., and Madnick, S. E. (2008) Measuring Data Believability: A Provenance Approach, Working papers 40086, Massachusetts Institute of Technology (MIT), Sloan School of Management.
- Redman, T. C. (2001) *Data Quality – The Field Guide*. Digital Press, p. 241
- SEI (Software Engineering Institute) (2006) “CMM for Development”, Version 1.2. Carnegie Mellon, Pittsburgh.
- Wand, Y. and Wang, R.Y. (1996) Anchoring Data Quality Dimensions in Ontological Foundation. *Communications of the ACM* 39 (11):86-95.
- Widom, J. (2005) Trio: A system for integrated management of data, accuracy, and lineage. *CIDR*, pp 262-276
- Witten, Ian H. And Frank, E., *Data mining: Practical Machine Learning Tools and Techniques*. Elsevier, San Francisco, Ca, 2005.