# Managing Software Quality through a Hybrid Defect Content and Effectiveness Model

Michael Kläs, Frank Elberzhager
Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany
{michael.klaes, frank.elberzhager}@iese.fhg.de

Haruka Nakao
Japan Manned Space Systems Corporation
Tsuchiura, Japan
haruka@jamss.co.jp

## ABSTRACT

Quality assurance (QA) plays a crucial role in today's software development. However, methods and models proposed in literature to support QA management suffer from several drawbacks. Many are specialized to certain activities like system test or inspections. They commonly support only one application purpose, e.g., planning or controlling, and are often applicable only after measurement data has been collected for several historical applications. To overcome these drawbacks, we developed a method that can be applied to QA activities during any phase, and which supports comprehensive quality management related tasks: improvement, planning, and controlling. To be applicable in practice, the method combines the available measurement data with expert judgment to build context-specific models. In addition, the method provides early benefits, while motivating the collection of measurement data by presenting possible improvement directions. The paper presents the general concepts behind the method and research questions to be answered in upcoming empirical studies.

## Categories and Subject Descriptors

D.2.9 [**Management**]: Software quality assurance (SQA)

## General Terms

Management, Measurement, Reliability, Experimentation

## 1. INTRODUCTION

Quality assurance (QA) is an essential part of today's software projects. The reduction of quality risk achieved by performing QA activities usually consumes a large portion of the project budget (between 30 and 90 percent) [1]. Therefore, managing software quality during the software lifecycle by identifying the major factors affecting the defect content and effectiveness of QA activities, as well as planning and controlling QA activities can contribute significantly to a project's success in terms of quality and project cost.

Managing software quality comprises different challenges in the software lifecycle, such as analyzing, planning, or controlling QA activities. Furthermore, prediction of the effectiveness of a QA activity can significantly improve the overall quality of the final product by choosing the best fitting QA activities, while prediction of the expected defect content can lead to a better decision regarding, e.g., the effort required for quality assurance.

However, predicting the effectiveness of a planned QA activity or the defect content of an artifact are no trivial tasks. On the one hand, effectiveness and defect content are both very context-specific; on the other hand, they are influenced by various factors (e.g., [2] identified more than 100 different factors with an impact on defect content or QA effectiveness).

Off-the-shelf methods like COQUALMO [3] use a smaller, fixed set of influencing factors to predict final defect content and overall effectiveness. However, [4] states that COQUALMO uses only coarse-grained categories of defect detection techniques, meaning that factors are not context-specific, and therefore do not fit in an optimal way. Moreover, from our practical experience, some of the COQUALMO factors are not relevant in every context and several context-specific ones are missing.

Furthermore, in order to manage different aspects regarding software quality, often more than one model type is necessary to draw all needed conclusions. Various models for different phases and purposes exist. For example, to cover the complete development cycle in terms of controlling QA, methods for static quality assurance (e.g., capture-recapture for inspection [6]) and testing (e.g. reliability growth models (RGM) [5]) have to be considered. But several models are not only limited in their application regarding different software development phases, moreover, different models are usually necessary to fulfill different purposes during one development phase. Examples with respect to inspections are MARS [7], which is used for planning, and capture-recapture models [6], which are used for controlling.

For valid conclusions to be drawn by models such as MARS [7], a suitable set of data is necessary. Thus, application is only possible if the high amount of required measurement data has been gathered for several projects, leading to bad motivation for starting data collection.

To overcome the above-mentioned problems, we propose a method that combines expert knowledge with measurement data to manage software quality. Depending on the available data, the built model can be used for holistically managing software quality purposes, such as identification of improvement potential, as well as planning and controlling of QA activities. From our experience, the typical situation in industry is that only few historical measurement data is available. Thus, the proposed method allows early benefits even if only few data is available. Furthermore, data collection is motivated by different application possibilities offered by the method if more data is gathered (e.g., application for the purposes of QA planning or QA controlling).

## 2. HDCE METHOD

The Hybrid Defect Content and Effectiveness method (HDCE) combines expert judgment and available measurement

data from current and historical projects to provide guidance for managing software quality. The idea of combining expert opinion and measurement data supported by a quantified causal model and Monte Carlo simulation is taken from the cost estimation area [8] and adapted to the quality assurance context.

Figure 1 provides a general overview of the relationship between the components of the HDCE method: The *quantified causal model* captures the expert opinions about relevant factors influencing defect content and effectiveness and their relative impact in the considered context. *Historical project data* are used to derive a defect content and effectiveness baseline for the context. Finally, the expert-based *characterization of the actual project* allows determining the *relative defect content and effectiveness probability distributions* (relative to other projects in the context). Monte Carlo simulation is applied to combine project characterization and the quantified causal model.

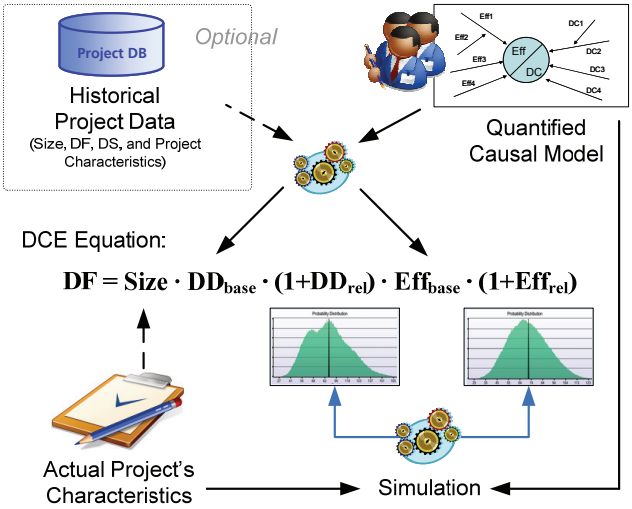


**Figure 1. Overview of the HDCE method**

The *DCE equation* is derived from the effectiveness definition in [9]: The effectiveness (*Eff*) of a QA activity (QA-A) is equal to the number of defects found (*DF*) by the QA-A divided by the defect content (*DC*) of the checked artifact before QA-A was performed. In addition, based on empirical evidence, the model assumes that the defect content has a linear dependency on the size of the artifact (*size*). Therefore, DC is split into size multiplied by defect density (*DD*). Finally, we expect that the defect density as well as the effectiveness of an actual project can be decomposed into a context-specific base value ($DD_{base}$, $Eff_{base}$) and a project-specific relative difference to this base value ($DD_{rel}$, $Eff_{rel}$) caused by the characteristics of the actual project (i.e., the actual impact of the factors influencing DC an Eff).

The process for building the quantified causal model is independent of the modeled QA-A, but the resulting causal model is context- and QA-A-specific, capturing the knowledge of local domain experts. Furthermore, the method supports different levels of application purposes (see Table 1). Whether and which historical project data are required depends on the purpose of application. Beginning with the purpose of QA explanation / improvement, which requires no historical data, the kind of data set required increases from lower-level to higher-level purposes. The data collected for lower-level purposes can be reused for applying the method to higher-level purposes, because the required data stack upon each other (Table 1).

Two independent improvement directions for organizations applying the HDCE method exist: On the one hand improvement along the purpose direction (low to high), realized by collecting certain additional measurement data over several projects (see requirements in Table 1), and on the other hand the application of the method for additional QA-A.

**Table 1. HDCE purposes, requirements, and outputs**

| ID: Purpose | Requirements* | Output |
|---|---|---|
| **1: QA Explanation / Improvement** | Quantified causal model for QA-A; Characterization of actual project | Pareto chart identifying DC and Eff influencing factors in actual project with the highest improvement potential |
| **2: Qualitative QA Planning** | (1) + size of checked artifact and characterization for >4 historical projects | Benchmarking of relative QA effectiveness and defect content of actual project against historical ones to identify projects with high quality risk |
| **3: (Quant.) QA Controlling** | (2) + number of defects found (DF) by QA-A for the historical projects | Thresholds for defects found by QA-A in actual project (based on DF probability distribution) |
| **4: Quantitative QA Planning** | (3) + number of defects slipped (DS) through the QA-A for the hist. projects | Prediction of absolute Eff and DC values for the actual project (i.e., actual DS can be predicted) |

* Required measurement data are underlined.

Figure 2 shows an improvement profile for an organization that performs four kinds of QA-A in their projects (RR, DR, CT, ST) and has built *quantified causal models* for three of them. For RR, they have collected the *number of defects found* in RR and the requirement *size* for several historical projects. Thus, they can use the method to control their RR by predicting thresholds for the number of defects expected to be found in RR. Since for ST, they also have the number of *defects slipped* the test, which are the defects found in the field, they can predict the absolute number of defects slipping actual ST based on ST-related project characteristics and the size of the artifact. However, since neither the number of defects detected in nor the number of defects slipped through CT are collected, the model can only be used for qualitative planning and improvement of CT activities.

If an organization reaches purpose level four for all their QA-A, they can plan and control their overall QA strategy for a project based on the defect content and effectiveness predictions available for each QA-A. The resulting defect injection and detection can be presented, e.g., by a defect flow model [10].

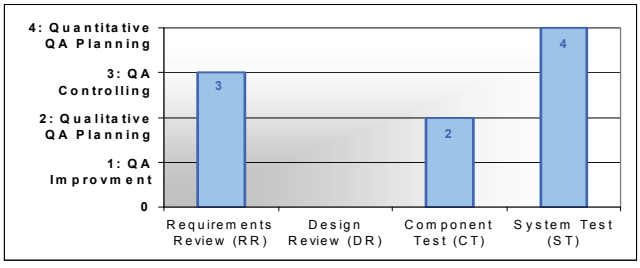


**Figure 2. Visualization of application phase x purpose**

As a third improvement direction, the improvement of model accuracy can be considered. Here, the availability of valid measurement data and the continuous maintenance of the quantified causal model play major roles.

## 3. METHOD EVALUATION

In order to evaluate the proposed hybrid defect content and effectiveness method, the applicability of the method and the usefulness of the results for the presented application purposes have to be investigated in empirical studies. In the following, we list open research questions followed by related hypotheses.

**Applicability:** The applicability of the method for a purpose depends mainly on the availability of the required measurement data (Table 1) and a reasonable quantified causal model:

**A1** Is it possible to build a reasonable, context-specific, quantified causal model together with local experts?

- The involved local experts reach agreement about relevant factors, i.e., a set of most relevant factors can be identified.
- The set of identified factors differs from the set of factors in general models like COQUALMO [3]. *Note:* It is only reasonable to build a context-specific model if the identified factors differ from factors in context-independent models.
- The identified factors explain the difference in defect content and effectiveness between different projects in the context.

**A2** How much expert involvement is required to build a quantified causal model for defect content and QA effectiveness?

- Number of local experts must not to exceed five.
- A quantified causal model can be built with less than two person-days of effort per local expert.

**Usefulness:** The usefulness of the method's results (Table 1) for the proposed application purposes can be evaluated partially by expert judgment, analysis of prediction accuracy, and comparison with the existing method that focuses on the considered purpose.

**U1** Is a Pareto chart with the impact of the different defect content and effectiveness factors useful for identifying the factors with the highest improvement potentials in the actual project?

- Experts agree on the model-based selection of factors with the highest improvement potential.

**U2** Does the method provide valid results for the purpose of quality risk assessment when planning QA activities?

- If experts are asked to rate historical projects with respect to quality risk, they arrive at a similar result as the method.
- Projects with higher quality risk predicted by the method have a higher number of defects slipping the QA-A.

**U3** How useful is the model for predicting the number of defects found by a QA activity?

- Experts agree that the provided DF (defects found) estimates are sufficiently accurate to support QA controlling.
- The estimation error of the method is significantly lower than the estimation error of applicable methods based solely on data or experts.

**U4** How useful are the absolute defect content and effectiveness predictions provided by the model?

- Experts agree that the DC and Eff estimates are sufficiently consistent and rationalized to support QA planning.
- The estimation error of the method is significantly lower than the estimation error of applicable methods based solely on data, or experts, or that of context-independent models.

## 4. CONCLUSION AND FUTURE WORK

In this paper, we briefly described the hybrid defect content and effectiveness method (HDCE). Different purposes, such as identifying improvement potential and planning and controlling QA activities are addressed by building context-specific models that consider the most relevant factors influencing both, defect content and QA effectiveness. The HDCE method combines available historical project data and expert judgment encapsulated in a reusable quantified causal model for factors influencing defect content and effectiveness. The method does not require the prior collection of massive amounts of historical measurement data for application, but provides some early benefits even if no or few measurement data is available. Furthermore, it motivates the collection of additional data in order to apply the model for advanced purposes later on.

At the moment, we are evaluating the proposed method in the field of high-dependability software development [11]. Our focus is the requirements phase and we have built a model for qualitative QA planning and controlling (U2 and U3). First outcomes show promising and valid results in terms of model accuracy, e.g., the predictions of the HDCE models are significantly more accurate than the predictions of models using only measurement data. A second evaluation will start soon in a different environment (i.e., different domain, later development phase). There, we want to apply all described purposes of the model to be built to generate holistic conclusions (U1-U4).

## 5. ACKNOWLEDGMENTS

Parts of this work have been funded by the BMBF SE2006 project TestBalance (grant 01 IS F08 D).

## 6. REFERENCES

[1] NIST: *The economic impacts of inadequate infrastructure for software quality*, 2002.

[2] J. Jacobs, J.van Moll, R. Kusters, J. Trienekens, A. Brombacher, *Identification of factors that influence defect injection and detection in development of software intensive products*. Inf. Softw. Technol. vol 49, Elsevier, 2006.

[3] S. Chulani, B. Boehm, *Modeling software defect introduction and removal: COQUALMO*, University of Southern California Center for Software Engineering, USC-CSE Technical Report 99-510, 1999.

[4] S. Wagner, *A Model and Sensitivity Analysis of the Quality Economics of Defect Detection Techniques*, ISSTA, 2006.

[5] M.R. Lyu, *Encyclopedia of Software Engineering. John Wiley & Sons, chapter Software Reliabiliy Theory*, 2002.

[6] L. Briand, K. El Emam, B. Freimut, O. Laitenberger, *Quantitative evaluation of capture-recapture models to control software inspections*. 8th International Symposium on Software Reliability Engineering, p. 234–244, 1997.

[7] L. Briand, B. Freimut, F. Vollei, *Using multiple adaptive regression splines to support decision making in code inspections*. Journal of Systems and Software, vol 73, 2004.

[8] L. Briand, K. El Emam, F. Bomarius, *COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment*, ISERN-97-24, 1998.

[9] S.H. Kan, *Metrics and models in software quality engineering*. 2. ed. Boston: Addison-Wesley, 2003.

[10] B. Freimut, C. Denger, M. Ketterer, *An industrial case study of implementing and validating defect classification for process improvement and quality management*. 11th IEEE International Software Metrics Symposium, 2005.

[11] M. Kläs, H. Nakao, F. Elberzhager, J. Münch, *Predicting Defect Content and Quality Assurance Effectiveness by Combining Expert Judgment and Defect Data – A Case Study*. Accepted at 19th IEEE International Symposium on Software Reliability, Nov. 2008.