Empir Software Eng (2010) 15:423–454
DOI 10.1007/s10664-009-9112-1

Support planning and controlling of early quality assurance by combining expert judgment and defect data—a case study

Michael Kläs · Haruka Nakao · Frank Elberzhager · Jürgen Münch

Published online: 11 July 2009

Editor: Laurie Williams

Abstract Planning quality assurance (QA) activities in a systematic way and controlling their execution are challenging tasks for companies that develop software or software-intensive systems. Both require estimation capabilities regarding the effectiveness of the applied QA techniques and the defect content of the checked artifacts. Existing approaches for these purposes need extensive measurement data from historical projects. Due to the fact that many companies do not collect enough data for applying these approaches (especially for the early project lifecycle), they typically base their QA planning and controlling solely on expert opinion. This article presents a hybrid method combining commonly available measurement data and context-specific expert knowledge. To evaluate the method's applicability and usefulness, we conducted a case study in the context of independent verification and validation activities for critical software in the space domain. A hybrid defect content and effectiveness model was developed for the software requirements analysis phase and evaluated with available legacy data. One major result is that the hybrid model provides improved estimation accuracy when compared to applicable models based solely on data. The mean magnitude of relative error (MMRE) determined by cross-validation is 29.6% compared to 76.5% obtained by the most accurate data-based model.

Keywords Software quality assurance · Quality management · Quality assurance effectiveness · Defect content estimation · Hybrid prediction model

M. Kläs (✉) · F. Elberzhager · J. Münch
Fraunhofer Institute for Experimental Software Engineering, Fraunhofer-Platz 1,
67663 Kaiserslautern, Germany
e-mail: michael.klaes@iese.fraunhofer.de

F. Elberzhager
e-mail: frank.elberzhager@iese.fraunhofer.de

J. Münch
e-mail: juegen.muench@iese.fraunhofer.de

H. Nakao
Safety & Product Assurance Department, Japan Manned Space Systems Corporation,
Urban Square Tsuchiura, 1-1-26, Kawaguchi, Tsuchiura, Ibaraki 300-0033, Japan
e-mail: haruka@jamss.co.jp

1 Introduction

Quality assurance (QA) is an essential part of today's software development projects. This holds in particular when dependable software-intensive systems are being developed, where delivering high quality is a major success factor. The reduction of quality risk achieved by performing QA activities such as reviews, inspections, model checking, code analysis, and testing usually consumes a large portion of the project budget (between 30% and 90%) (NIST 2002). Therefore, the accurate planning and controlling of QA activities (in particular QA activities during early software lifecycle phases like requirements reviews and inspections) can contribute significantly to the project's success in terms of quality and project cost (Shull et al. 2002). When planning QA activities in a project, one major question is whether the planned QA activities are appropriate for reducing the product-specific quality risk to an acceptable level, i.e., whether the planned activities are effective enough to handle the expected defect content of the checked product. The next issue is to determine whether the planned level of risk reduction has been achieved. Thus, indicators or thresholds for the expected number of defects found by the planned QA activities are required.

However, predicting the effectiveness of a planned QA activity or the defect content of an artifact is not a trivial task. Earlier research focused on finding a single size or complexity measure as an accurate predictor of defect content (Halstead 1977), (McCabe 1976). Another approach was to provide an effectiveness value or range for certain QA techniques by aggregating the collected measurements from different companies (Jones 1996) or the results from different empirical studies. In practice, both led to disappointment: Single size and complexity metrics are recognized as being insufficient predictors of defect content (Fenton and Neil 1999) and the effectiveness of QA techniques varies extensively depending on the concrete context of the study (Aurum et al. 2002, Juristo et al. 2002). Therefore, effectiveness values described in the literature cannot be applied directly as predictors of QA effectiveness in an organization's own context.

From our point of view, the reason for the observed problems is that the effectiveness of QA activities as well as the defect content of checked artifacts are complex constructs, which are (1) very context-specific and (2) influenced by several factors. A reasoned literature research (Jacobs et al. 2007) identified over 100 different factors mentioned across the software engineering literature that have an impact on defect content or QA effectiveness.

Newer research directions like Briand and Freimunt (2004) and Bibi et al. (2006) are tackling the problem by using multivariate approaches, which allow building context-specific models and consider more than one independent variable (i.e., influencing factor). These approaches are applied successfully in areas where large datasets are available (more than 100 data points) or where data collection is possible without major effort, respectively. For example, Briand et al. (2000a) and Nagappan et al. (2006) used automatically collected code metrics to successfully predict defect-prone modules.

However, practical application of these models is limited by their data requirements, which holds especially for early software development phases, where no automated data collection is available. For example, only few companies have access to collected measurement data about the effectiveness of inspections and relevant influencing factors for more than 100 requirements inspections (i.e., historical data points).

Consequently, we see a lack of empirically validated methods that can be used to build context-specific prediction models for defect content and effectiveness when limited quantitative data is available.

Considering the field of cost estimation, methods are described for building and validating context-specific prediction models, even in cases where limited measurement data from only few historical projects are available. Methods like CoBRA® (Briand et al. 1998) have been applied successfully in many case studies to build context-specific prediction models using expert experience and existing measurement data (Briand et al. 1998, Ruhe et al. 2003, Trendowicz et al. 2006). Therefore, we adopt the idea of combining expert opinion with measurement data in a rigorous manner and adapt it for defect content and QA effectiveness prediction. The method we present supports the planning of early QA activities by providing assessments of the remaining quality risks related to the planned QA activities and providing thresholds for controlling the planned QA activities.

To evaluate the proposed method with respect to applicability, application cost, and usefulness, a case study was performed. The study was conducted in the context of software Independent Verification and Validation (IV&V) performed for mission-critical space systems. Independence is considered here in terms of technical, managerial, and financial independence (IEEE 2005).

The paper is organized as follows: Section 2 discusses related work regarding planning and controlling early quality assurance techniques. Section 3 presents the Hybrid Defect Content and Effectiveness Early Prediction method (HyDEEP), which allows building a hybrid prediction model combining expert opinion and measurement data for planning and controlling QA activities. In Section 4, the method is evaluated in a case study with respect to applicability and application cost. A set of context-specific influencing factors is presented, and the usefulness of the resulting model is evaluated by applying the model on legacy data. Finally, Section 5 concludes the paper and outlines directions for future research.

2 Related Work

While considering prediction models for the planning and controlling of quality assurance techniques applied during early lifecycle phases, we identified various techniques. One approach to *controlling inspections* is the use of capture-recapture models (Briand et al. 1997), which were originally used in the domain of biology. They answer the question of whether inspections can be stopped or should be continued in terms of the number of defects remaining within a document. For this purpose, an estimate of the total number of defects within a document is based on the defects already found by different inspectors. A calculation of the effectiveness and the estimated number of remaining defects can be done based on the assumption that a large number of defects found by only one inspector results in a high overall defect count. On the one hand, if many defects are found by more than one inspector, the overall number of defects is estimated to be low. On the other hand, if many defects are found by only one inspector, the number of undetected defects is estimated to be high (Eick et al. 1992). The main problem is the number of inspectors required to obtain suitable estimation results, which is a minimum of four (Petersson et al. 2004). Moreover, most estimations underestimate the number of remaining defects (Petersson et al. 2004).

Another approach to controlling inspections are curve-fitting models as described by Wohlin and Runeson (1998), which are derived from the idea of reliability growth models used for controlling testing activities. Here, defect data is gathered for inspection activities and different distributions are calculated to predict the overall number of defects. Although different models and improvements for such models have been developed, the accuracy of curve-fitting models was not as high as that of capture-recapture methods (Petersson et al. 2004).

Besides these model-based approaches other techniques are also used to control early QA activities. Strictly data-based ones come from the area of statistical process control (Kan 2003). They are applied to determine a typical range (called control limits) for the expected number of defects found by the QA activity based on the number of defects found in earlier applications (Weller 1994). Successfully applied in production industry, where they originate, their application in software QA is more restricted due to the fact that the number of data points is more limited and that the processes in software engineering are less repeatable than in production industry. As a result, the calculated control limits can become too broad to be useful for controlling QA. In addition, due to the assumption of a stable process, factors influencing the defect content and effectiveness are not considered. As a result, a good product containing only few defects may be inspected twice since too few defects are found, and a product containing many defects, of which an expected number are found in an inspection not performed that well might not be inspected again.

Finally, there are also expert-based early QA controlling approaches, which are completely independent of historical data (El Emam et al. 2000). Typically, they ask the experts after the inspection for the expected number of remaining defects, letting them implicitly estimate the defect content in the beginning of the inspection and their effectiveness in finding defects. However, these approaches suffer from the typical limitations of expert estimates, such as the limited repeatability of the results, limited availability of experienced experts, and many sources of bias such as social or group pressure (Meyer and Booker 2001). Because estimates are provided only by those experts who performed a particular inspection, these biases are difficult to handle and are especially problematic.

For the purpose of *planning inspections*, only few prediction models exist. Previous research mainly focused on controlling inspections. One approach to planning inspections is described by Briand et al. (2000b). Data gathered from about 150 inspections are used to perform a linear regression analysis, considering the size of the artifact and the preparation effort. The main problems with this approach is the extensive number of data points necessary for performing such an analysis and the fact that only the impact of a very limited number of factors is considered. A model that uses only two factors, document size and preparation effort, may provide limited predictive power if further factors (e.g., domain experience of the inspectors or the problem complexity of the inspected artifact) vary in the application context and affect the inspection results. Finally, MARS (Multiple-Adaptive-Regressions-Splines) (Friedman 1991) has shown its suitability for planning compared to different regression analyses, based on three studies. However, MARS also requires many data points (>100 inspections) for valuable application (Freimut 2006).

Consequently, the main problem regarding existing methods is the availability of information sources to base estimation on. On the one hand, there is the need for a large number of inspectors for suitable predictions and, on the other hand, a large amount of necessary data must be gathered for inspections, which is often not available.

3 The HyDEEP Method

The HyDEEP method combines expert judgment and available measurement data from current and historical projects to provide guidance for QA planning and controlling. The idea of combining expert opinion and measurement data supported by a quantitative causal model and Monte Carlo simulation is taken from the cost estimation area (Briand et al. 1998) and adapted to defect content and effectiveness prediction.

The *Hybrid Defect Content and Effectiveness* (*HDCE*) *model* is the core element of the HyDEEP method. Figure 1 provides a general overview of the relationship between its components: The *defect content and effectiveness (DCE) equation* combines the *defect content* and the *effectiveness model*; the *quantitative DCE causal model* captures the expert opinions; the *historical project data* are used to derive a defect content and effectiveness baseline for the application context; and the *characterization of the actual project* allows determining the project-specific *defect density increase factor (DDIF) and effectiveness improvement factor (EIF) distributions* with the help of *Monte Carlo simulation* (Fishman 1995).

In the following subsections, we first describe the main components of the HDCE model: the DCE equation (Section 3.1) and the quantified DCE causal model (Section 3.2). Then we give an overview of the process employed for building the quantitative causal model using expert judgment (Section 3.3) and how the DDIF and EIF distribution can be determined by Monte Carlo simulation (Section 3.4). Finally, we explain which historical project data are required (Section 3.5) and how the HDCE model can be applied to QA planning and controlling (Section 3.6).

3.1 Defect Content and Effectiveness Equation

The number of *defects found* (DF) by a QA activity can be described by two variables, the number of defects in the checked artifact at the moment when the QA activity was

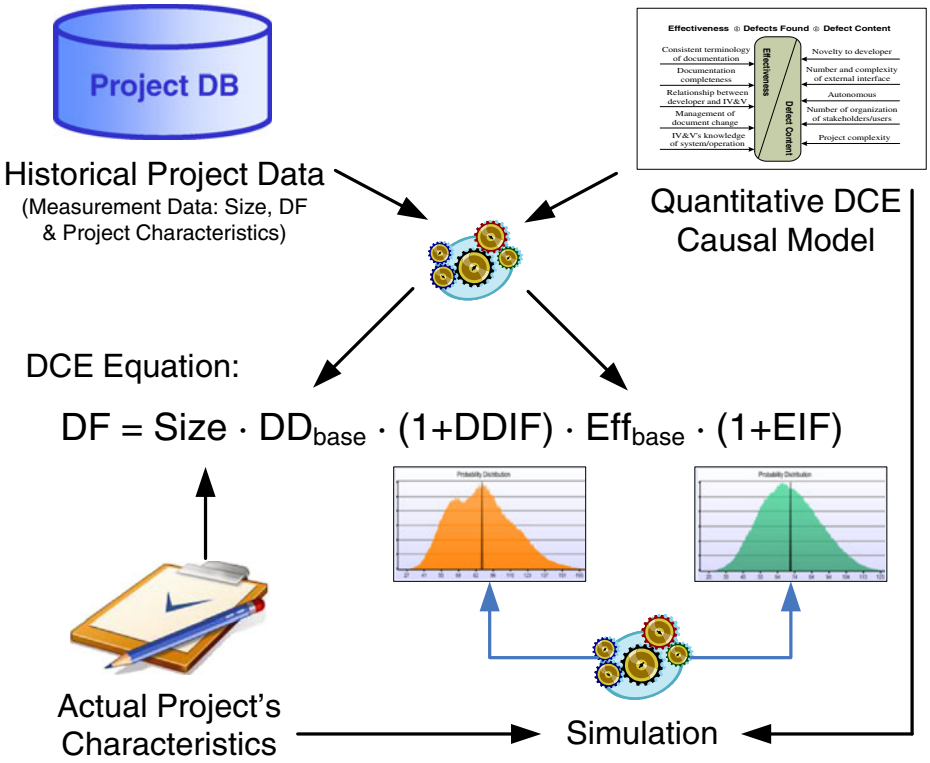


Fig. 1 In the HDCE model, data and expert judgment are combined by simulation

performed (i.e., the defect content) and the effectiveness of the QA activity itself, meaning the ratio between the number of defects found and the total number of defects in the product when the QA activity is conducted (Kan 2003). In the following, we describe one model for defect content and one model for QA effectiveness. Equation (1) describes how these models are related and Fig. 1 shows the resulting DCE equation as the glue between the other parts of the HDCE model:

$$defects\,found = defect\,content \cdot effectiveness \tag{1}$$

Defect Content Model In contexts where a well-established development process is used and the development team is homogenous and stable (i.e., similar software units are developed or maintained over years), the major factor determining *defect content* (i.e., the number of defects inside the artifact when the QA activity starts) is the size of the artifact (Endres and Rombach 2003). In such a context, we get a stable *defect density* (DD) value over our projects:

$$defect\,density = defect\,content / size \tag{2}$$

In most environments, however, more factors exist besides artifact size that influence the defect content (e.g., developer experience, novelty of application, time pressure). The DC model captures the accumulated influence of these factors as the *defect density increase factor* (*DDIF*). Thus, defect density can be split into a *base defect density* (DD_{base}), which represents the hypothetical minimum defect density reachable in the considered context (best case), and a *defect density increase factor* (*DDIF*) describing the relative increase of this base defect density caused by the influence of factors increasing the base defect density:

$$defect\,content = size \cdot DD_{base} \cdot (1 + DDIF) \tag{3}$$

For illustrating (3), imagine artifacts produced in projects A and B have the same size, but project A has a *DDIF* value of 0.0 (best case) and project B has a *DDIF* value of 0.3: then the artifact checked in project B has a 30% higher defect content $\left(DC_{\mathrm{Pr}ojectB} = 1.3 \cdot DC_{\mathrm{Pr}ojectA}\right)$.

Effectiveness Model The defect removal *effectiveness* of a QA activity can be defined as the number of defects found by the QA activity divided by the total number of defects in the product before the QA activity starts (Kan 2003). Like the defect density, the effectiveness of a QA activity depends on different influencing factors (e.g., experience of the QA team, understandability of the checked artifact, available tools). The accumulated impact of these factors on the effectiveness of a QA activity in a concrete project is captured by the *effectiveness improvement factor* (*EIF*). Therefore, like the actual defect density, the actual effectiveness can be split into two components: The *base effectiveness* (Eff_{base}) represents the hypothetical minimum effectiveness of the QA activity in the considered context (worst case), and the *effectiveness improvement factor* (*EIF*) describes the relative increase of this base effectiveness caused by the influence of effectiveness improvement factors.

$$effectiveness = Eff_{base} \cdot (1 + EIF) \tag{4}$$

For illustrating (4), imagine project A has an *EIF* value of 0.0 and project B has an *EIF* value of 0.2: then the QA activity in project B is 20% more effective than the activity in project A $\left(Eff_{\mathrm{Pr}ojectB} = 1.2 \cdot Eff_{\mathrm{Pr}ojectA}\right)$.

3.2 Quantitative DCE Causal Model

In general, providing estimates of the defect density increase factor (*DDIF*) and the effectiveness improvement factor (*EIF*) is a difficult task. We propose determining the defect density increase factor and the effectiveness improvement factor by building and applying a *quantitative causal model* for defect content and effectiveness.

Such a model consists of three components. First, it contains the most important factors influencing *defect density* and *effectiveness* and their causal relationship in the considered context (example: Fig. 2). Second, for each influencing factor, it provides a scale with usually four levels, which allows experts to characterize a project with respect to the factor. Finally, for each factor, it captures the experts' judgment about the factor's impact on the number of defects detected in a quantitative way. The process we propose in the next section for building such a model is similar to the process successfully applied in the cost estimation method CoBRA® (Briand et al. 1998) for building quantitative causal models for productivity.

3.3 Process: Quantitative DCE Causal Model Building

The quantitative causal model captures the expert-opinion-based part of the HDCE model. This section gives a short overview of the process proposed for building such a quantitative causal model for defect content and effectiveness in a given context. The process consists of five major steps: (1) Define the application context and collect relevant influencing factors, (2) identify the most important ones, (3) build a causal model based on these factors, (4) define a rating scale for each factor in order to characterize projects with respect to the factor, and, finally, (5) quantify the impact of each factor on defect content or QA effectiveness based on expert judgment.

3.3.1 Collecting Relevant Influencing Factors

When considering defect density, usually some factors other than size exist that influence the number of defects in a product (e.g., developer experience, novelty of application, time pressure). The same holds for QA effectiveness. Factors of the first type will affect *DDIF*; factors of the second type will affect *EIF*. The relevant factors depend on the context in

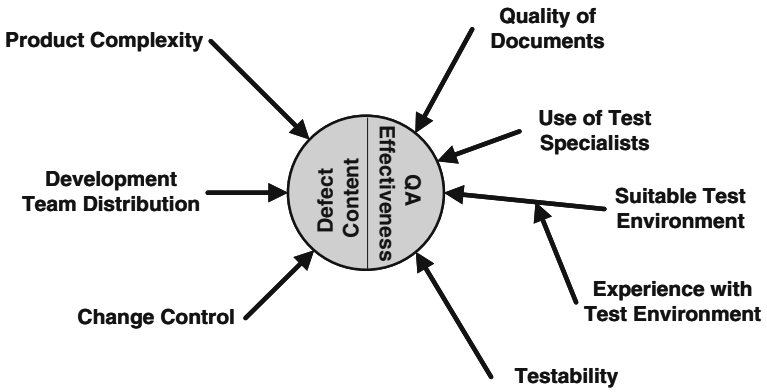


Fig. 2 Example of a defect content and effectiveness causal model

which the model should be applied and which should be defined before. Based on the context definition, the domain experts collect a list of relevant factors during a brainstorming session. *Relevant* means (1) the level of the factor varies across projects, (2) the level of the factor can be determined or at least reasonably judged for each project, and (3) the experts assume that the variation has a noticeable impact on defect content or QA effectiveness.

The factor identification process should be supported by a list of factors that may be relevant in the context. Such a list can be extracted from the literature (e.g., Jacobs et al. (2007)) or reused from an earlier application in a similar context. When the brainstorming session is finished, the list should be used to check completeness together with the experts (to ensure that no relevant factor is missed).

In addition to the *factor name* and a *short description*, a context-specific realistic *worst case* and *best case* should be recorded for each factor. Recording a context-specific best and worst case has two advantages: First, if no best and worst case can be provided, this is typically an indication that the factor levels do not vary in the context or cannot be determined. Second, providing a context-specific variation range for each factor simplifies the identification of the most important factors in the subsequent step.

3.3.2 Identifying the Most Important Influencing Factors

After elicitation of all relevant factors, the factors are ranked by the experts regarding their importance. A factor is considered more *important* than another if its variation is assumed to be responsible for a higher variation in the number of defects detected by QA than the factor compared.

Usually, the most convenient way to get this ranking is a questionnaire where factors are separated into groups of six to 12 each and each expert is asked to provide a ranking from one to n with respect to factor importance for the factors in each group. Groups with more than 12 factors are very difficult to rank for experts, since too many comparisons have to be performed. For example, experts have to perform up to 105 comparisons to rank a list with 15 factors but only 30 to rank three lists with five factors each. The rankings provided by the experts can be analyzed with the help of descriptive statistics (e.g., mean, min, max, standard derivation) and Kendall's coefficient of concordance (Kendall and Smith 1939) in order to identify the most important factors and determine the agreement between the experts regarding the importance of certain factors.

3.3.3 Building the Causal Model

In general, two options are possible for building the causal model based on the analysis results. The first one, the *rigorous-ranking-based* option, is to build the model considering only the analysis results, i.e., selecting the most important factor in each category and all factors whose mean rank is at most 10% higher than the mean rank of the most important factor. In this case, no interactions between factors are considered. On the one hand, not considering interactions between factors naturally reduces the expressiveness and, therefore, the maximal obtainable accuracy of the model. On the other hand, including interactions between factors increases model complexity and makes quantification of the factors' impact more complicated for the experts, which again can reduce the accuracy of the final model. In the area of cost estimation, e.g., Ruhe et al. (2003) argues against using interactions and for working with a set of direct influencing factors instead. The second,

discussion-based option is to review the analysis results in a workshop with the experts in order to decide which factors should be included in the model. The second option provides the advantage that disagreements between experts with respect to the meaning or importance of factors can be articulated. However, this option typically requires significant logistical effort for the meeting and results in more complex causal models.

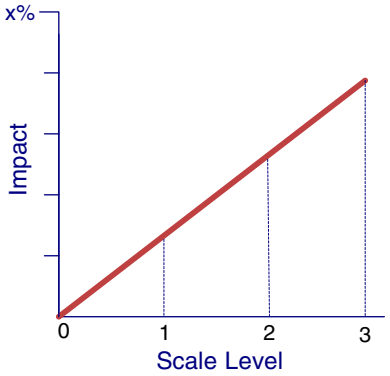
In any case, considering the experience with causal models in cost estimation (Trendowicz et al. 2006), four to six defect content and four to six effectiveness factors can be considered as a reasonable number of factors for a DCE causal model.

3.3.4 Defining Scales for the Influencing Factors

In order to characterize the value of an influencing factor for a project, scales with levels from zero to three are used. The description for level 0 should describe a realistic situation in the considered context where the factor leads to a minimal increase in the number of defects found; the description for level 3, on the other hand, should describe a realistic situation in the context where the factor results in a maximal increase in the number of defects found. *Note:* Based on this definition, level 3 is the *worst case* for defect content factors (high defect content) and the *best case* for effectiveness factors (high QA effectiveness). The descriptions for levels 1 and 2 should be defined in such a way that the impact of the factor on the number of defects found increases (if possible) in a linear manner from 0 via 1 and 2 to level 3 (see Fig. 3). Defining a context-specific and realistic best case and worst case is essential because in the next step, experts have to determine the quantitative impact of each factor by comparing the best case with the worst case.

3.3.5 Determining the Impact of Influencing Factors

The influence of a factor is defined as a relative percentage increase of defect content or QA effectiveness (i.e., the number of defects found by the QA activity). For each defect content factor, experts are asked to provide an estimate for the expected increase in defect content when the considered factor has the worst possible value (level 3) compared to the best case



D4: Number of stakeholder/user organizations	
Level	**Description**
0	Stakeholders are customer and supplier.
1	… customer, supplier, and user.
2	… customer, supplier, and international partner.
3	… customer, supplier, several users, and international partner.

Fig. 3 The impact of the influencing factor increases linearly over the factor's levels

(level 0). For each effectiveness factor, experts are asked to provide an estimate for the relative number of more defects found when the considered factor has the best possible value (level 3) compared to the worst case (level 0). Considering experts' uncertainty and the uncertainty of a factor's concrete impact in the context, they are not asked for a point estimate (i.e., one value) but for a *minimum*, *most likely*, and *maximum* estimate (see Fig. 4). These three values define a triangular probability distribution capturing the uncertainty of the estimate, a common technique in quantitative risk analysis (Vose 1996). If the DCE model contains interaction between influencing factors, the questionnaire for these factors become more complicated. We did not use interactions in the case study and recommend avoiding them if there are no good reasons for including. The use of interactions makes the questionnaire for impact elicitation and the simulation significantly more complex. For more details on two-way and three-way interactions in cost estimation causal models, see for example (Briand et al. 1998).

3.4 Determine DDIF and EIF Through Simulation

Since *DDIF* and *EIF* are determined based on expert judgment, which contains uncertainty, modeling *DDIF* and *EIF* as an exact value would be inappropriate. A single value cannot capture the uncertainty inherent in the usage of expert opinion. Therefore, *DDIF* and *EIF* are calculated and presented as probability distributions. For a project characterized by the level of each factor, Monte Carlo simulation (Fishman 1995) is applied to determine the *DDIF* and *EIF* probability distributions with the help of the quantitative causal model. The equations used to derive these probability distribution are described in this section.

In order to quantify the impact of each factor (f) in the DCE causal model (see Section 3.3.5), each expert e provided for each factor a minimum (*MIN*), most likely (*ML*), and maximum (*MAX*) estimate (see Fig. 4). This means for each expert and factor that a triangular probability distribution $triang_{e,f}(MIN_{e,f}, ML_{e,f}, MAX_{e,f})$ is defined, where the interval [*min,max*] defines the range and the *ml* value the peak of the triangular.

This distribution represents the estimate of the expert for the situation of maximal factor impact (i.e., if the factor has level 3). Therefore, the triangular distribution has to be

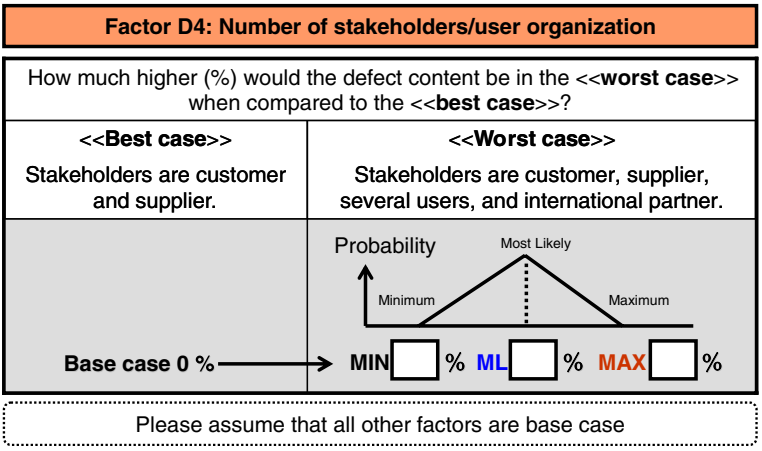


Fig. 4 Extract from a questionnaire used to determine the impact of the influencing factors

adjusted by the *factor level* ($level_{p,f}$) of the considered *project* (p). In accordance with the linearity of impact increase over factor levels (see Fig. 3), the project-specific triangular distribution is defined by (5):

$$triang_{e,f,p} = triang\left(\frac{level_{f,p}}{3} \cdot MIN_{e,f}, \frac{level_{f,p}}{3} \cdot ML_{e,f}, \frac{level_{f,p}}{3} \cdot MAX_{e,f}\right) \tag{5}$$

When we now consider the set of defect content factors (*DFs*) and effectiveness factors (*EFs*) in the causal model as well as the set of experts (E) that provided estimates for these factors, *DDIF* and *EIF* distributions are defined by (6):

$$EIF_p = \sum_{f \in EFs} Dist_f \quad DDIF_p = \sum_{f \in DFs} Dist_f \text{ with} \tag{6}$$

$$Dist_f = \begin{cases} triang_{e_1,f,p} \text{ with probability } \frac{1}{|E|} \\ triang_{e_2,f,p} \text{ with probability } \frac{1}{|E|} \\ \quad \ldots \\ triang_{e_{|E|},f,p} \text{ with probability } \frac{1}{|E|} \end{cases}$$

Using (6), the DDIF and EIF distributions can be determined by Monte Carlo simulation (Fishman 1995) or the more complex but also more efficient Latin Hypercube approach (McKay et al. 1979). More details on the application of Monte Carlo simulation to determine these kinds of distributions can be found, for example, in (Kläs et al. 2008), where an approach is also described to calculate the expectation value of these distributions analytically.

3.5 Historical Project Data

When the HDCE model is to be applied for QA planning and controlling, three kinds of data have to be collected for at least four to five historical projects in the context. The measurement data used in the model is limited to data typically collected in projects or easy to reconstruct from historical project documentation. The first kind of data used is the number of *defects found* (*DF*) by the considered QA activity. Here, it is important to have a clear definition of 'what is a defect' and that this definition is consistently used in the selected historical projects. Second, for the same historical projects, the *size* of the artifacts checked by the QA has to be determined. Here, the same applies as for the number of defects found. Finally, each historical project used in the model have to be characterized by determining the levels of the identified defect content and effectiveness factors. This characterization is not based on measurement data but on expert judgment using the previously defined scales for each factor (see Section 3.3.4). The characterization is usually provided by experts who have participated in the project or know the project very well.

3.6 HDCE Model Application

This section presents how the quantified causal model can be applied for QA planning and QA controlling. It describes the prerequisites for applying the HDCE model for a certain purpose, how the calculations are performed, what the results are, and how they can be used.

3.6.1 QA Planning: Evaluate Quality Risk

The quantified causal model can be used during the QA planning phase to evaluate the remaining quality risk based on the planned QA activities. The basic idea is to benchmark the current project against historical projects with respect to expected defect density and QA effectiveness.

The required information includes expert-based characterizations of the current and at least four to five historical projects with respect to the level of influencing factors defined in the quantitative causal model. The characterization of the projects allows calculating the expected *DDIF* and *EIF* values for each project by Monte Carlo simulation (see Section 3.4). In the following, we assume that effort spent on the QA activity is adapted to the size of the checked artifact (e.g., for each document page, two person-hours are available for the planned QA activity). If the effort spent per size unit varies across different projects, this fact can be considered as an factor influencing QA effectiveness (e.g., as the intensity of the QA activity). The result of the *DDIF* and *EIF* calculations is visualized in a Cartesian coordinate system where the zero point of the x-axis (named *relative defect density*) is equal to the mean *DDIF* over all *n* historical projects (7) and the zero point of the y-axis (named *relative effectiveness*) is equal to the mean *EIF* over all historical projects (8).

$$relative\,defect\,density = DDIF - \sum_{i=1}^{n} DDIF_i \tag{7}$$

$$relative\,effectiveness = EIF - \sum_{i=1}^{n} EIF_i \tag{8}$$

Based on the relative defect density and QA effectiveness of the actual project when compared to the historical projects, the actual project can be found in one of the four quadrants (Q1) to (Q4) of the coordinate system (see Fig. 5):

- *(Q1)* *High defect density and effectiveness* usually means no major quality risk, since the relative number of defects that slipped through the QA activity is low because of the high effectiveness.
- *(Q2)* *Low defect density but high effectiveness* usually means a very low quality risk, but can also mean an inappropriately high QA intensity with respect to the defect density expected (resulting in unnecessary costs).
- *(Q3)* *Low defect density and effectiveness* might also not mean major quality risks, since the relative number of defects that slipped through the QA activity is low (because of the low defect density).
- *(Q4)* *High defect density but low effectiveness* means a quality risk, since a relatively high number of defects can slip through the QA activity and result in potentially low product quality.

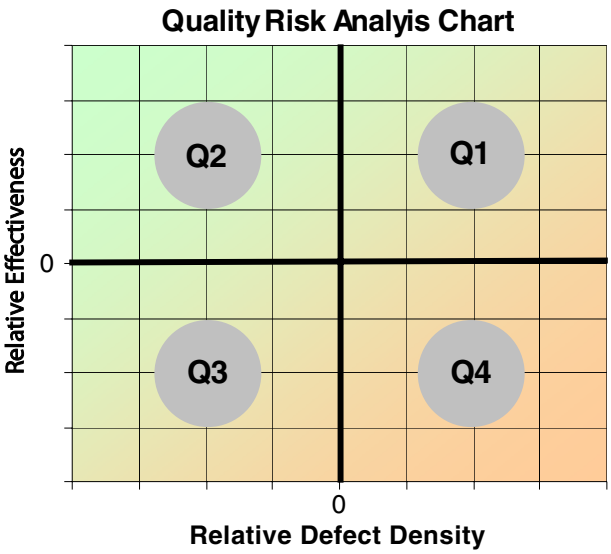


Fig. 5 The four quadrants in the quality risk chart

3.6.2 QA Controlling: Estimating Defects Found

Besides QA planning, the quantified causal model can be used to predict the expected number of defects found by QA activities. This information can be used to control the QA process, i.e., if significantly fewer or more defects are found, the reason must be identified. The reason may be a deviation from the expected defect density or a derivation from the planned effectiveness of the QA activity.

In the case of higher effectiveness or lower defect density, this is usually not a problem, but in the case of lower effectiveness or higher defect density, this may be a quality risk leading to low final product quality or high correction costs in later project phases. Therefore, countermeasures should be initiated to handle the identified higher risk.

Since for each historical project j, $DDIF_j$ and EIF_j can be calculated based on the characterization of the project with respect to the factors in the quantitative causal model (see Section 3.4), we can use these values together with the artifact size and the number of defects found to calculate the (DD_{base}·Eff_{base}) value of the project using (9):

$$(DD_{base} \cdot Eff_{base})_j = DF_j / \left(size_j \cdot \left(1 + DDIF_j\right) \cdot \left(1 + EIF_j\right)\right) \tag{9}$$

Based on our model assumptions, this value should have low variation across projects in the specified context because variations in defect density and effectiveness are encapsulated in the *DDIF* and *EIF* values. Therefore, the median of the n historical (DD_{base}·Eff_{base}) values is used to provide an estimate for the (DD_{base}·Eff_{base}) value of the actual project (10):

$$(DD_{base} \cdot Eff_{base})_{est} = Median\left(\left\{(DD_{base} \cdot Eff_{base})_j\right\}_{j=1..n}\right) \tag{10}$$

The median and not the mean is used because the median is less affected by outliers than the mean. If a sufficient number of historical projects is available for model building—

around ten projects—(robust) regression analysis can also be an alternative for determining the $(DD_{base} \cdot Eff_{base})_{est}$ value for the actual project.

When $(DD_{base} \cdot Eff_{base})_{est}$ has been determined, the expected number of defects found (DF_{est}) can be calculated by applying the DCE equation for the actual project (11):

$$DF_{est} = size \cdot (1 + DDIF) \cdot (1 + EIF) \cdot (DD_{base} \cdot Eff_{base})_{est} \quad (11)$$

4 Case Study

4.1 Context of the Study

The case study presented here was performed in the context of independent verification and validation (IV&V) of mission-critical, on-board space system software. The objective was to build a hybrid prediction model for the IV&V activities performed during the software requirements analysis phase (SRA). These SRA activities are mainly performed by document review. Depending on the project's situation, model checking is sometimes performed.

The data applied for constructing and evaluating the HDCE model came from five historical projects where software IV&V was performed during SRA. These projects were international collaborative projects (e.g., JEM and the HII-A Transfer Vehicle) and other spacecraft projects requiring an expense budget for IV&V (Kohtake et al. 2008).

Table 1 shows information about the domain experts who participated in building the model by answering the prepared questionnaires.

4.2 Overall Study Goals and Research Questions

The primary goal of the study was to evaluate the applicability of the HyDEEP method and the usefulness of the resulting model by answering the following research questions:

RQ1: Is the HyDEEP method, which combines expert and measurement data, applicable in the context of defect content and effectiveness prediction?

RQ2: How much expert involvement (in terms of effort) is required to build the quantitative DCE causal model?

Table 1 Involved domain experts

Expert ID	Role of the expert	Experience [#years]	
		Domain	IV&V
1	SPA/IV&V	20	2
2	SPA/IV&V	25	2
3	SW Safety/IV&V	17	10
4	IV&V	17	5
5	IV&V	13	9
6	IV&V	5	5
7	SW Safety/SPA/IV&V	8	8

SPA Software product assurance, *SW Safety* Software safety

RQ3: Does the model built provide plausible results for quality risk assessment when planning QA activities?

RQ4: How useful is the model for predicting the number of defects found by a QA activity?

Research questions RQ1 and RQ2 are answered in Section 4.3 by applying the proposed HyDEEP method to build a quantitative causal model for defect content and QA effectiveness. Afterwards, the resulting model is evaluated in Section 4.4 regarding its usefulness for QA planning and controlling (RQ3 and RQ4) using historical project data (Fig. 6).

4.3 Model Building: Evaluation of RQ1&RQ2

4.3.1 Study Objective and Design

The ability to build a context-specific causal model for defect content and effectiveness primarily depends on two prerequisites: (1) experts able to identify a set of factors they consider to have the highest impact in their context, and (2) a certain degree of agreement between the judgments of the domain experts.

Whether an identified factor really has the assumed impact in the considered context cannot be determined directly in the case study, but can only be measured indirectly by checking the improvement of estimation accuracy when applying the HDCE model containing the identified factors. The estimation accuracy will be evaluated in detail in Section 4.4 (RQ4) with the help of legacy data.

We follow the causal model building process (Section 3.3) and defined the following study objectives for RQ1:

Objective 1 After independent factor ranking by all involved experts, check which level of agreement experts reach regarding the most important factors.

Objective 2 Check whether a quantified DCE causal model can be built together with the local domain expert.

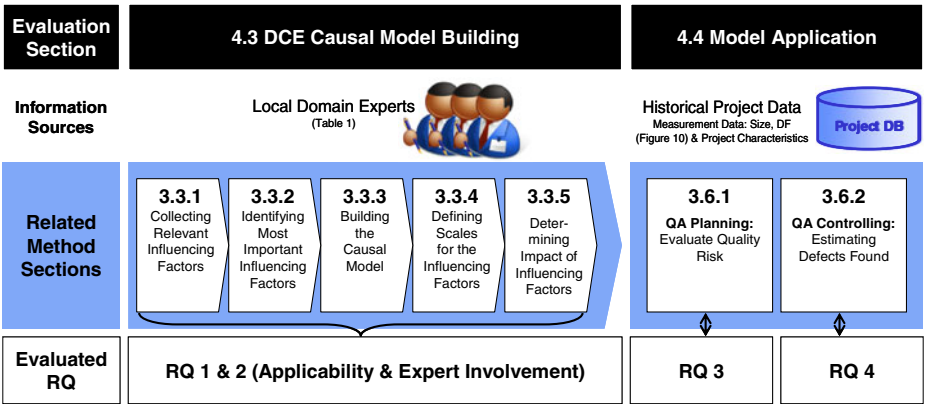


Fig. 6 Overview: Empirical evaluation process and research questions

Objective 3 Check whether the set of factors in the DCE causal model differs from the set of general defect introduction and removal factors in COQUALMO (Huang and Boehm 2005), since it is not reasonable to build a context-specific model if the identified factors do not differ from the factors in context-independent models.

In order to evaluate RQ2, the degree of expert involvement required is determined by collecting the effort data of the involved experts separately for each expert and activity.

4.3.2 Study Instrumentation and Execution

Identification of Relevant Factors In order to identify the influencing factors, we used results from historical discussions in the IV&V community (Nakao et al. 2007) as input. In addition, we included domain-related factors for defect content and effectiveness from the literature (Jacobs et al. 2007). Initially, we had a set of more than 100 identified factors. From this set, we carefully selected context-relevant factors by discussing the factors with a domain expert to decide whether (1) the level of the factor varies in the context, (2) the variation of the factor level has an impact on defect content or IV&V effectiveness, and (3) the level of the factor can be determined by IV&V personnel. Finally, we identified 21 relevant defect content (see Table 2) and 18 relevant effectiveness factors (see Table 3).

Identification of Most Important Factors In a first questionnaire, seven domain experts were asked to rank the factors with respect to their importance (see Section 3.3.2). In order to make the ranking easier for the experts, the factors were not only separated by the categories defect content and effectiveness, but also by the categories 'product', 'project', and 'process & personnel' (see Tables 2 & 3). This allowed the experts to rank the factors individually in each of the six categories. The mean of the expert-based rankings that a factor received is presented in the table, where lower rankings mean higher importance. The agreement between the experts is measured by *Kendall's coefficient of concordance* (W) (Kendall and Smith 1939). W ranges between zero and one, with one corresponding to perfect agreement between the experts and zero indicating no agreement or independence of the sample.

Factor Selection for the Causal Model Since no model building workshop with the domain experts could be performed, we decided to select factors based solely on statistics (see *rigorous-ranking-based* option described in Section 3.3.3). In Tables 2 and 3, the factors identified for inclusion in the model are highlighted.

Factor Scale Definition Based on the discussion between a domain expert and a measurement expert, the final scales for the factors were defined. For each factor, the scales provide a context-specific wording to define the factor's levels 0 to 3 (see Section 3.3.4). For level 0 and level 3, a concrete worst and best case situation was chosen based on the experience from historical projects in the context.

Factor Quantification with Questionnaire A second questionnaire supported the seven experts quantifying the impact of the factors on defect content and IV&V effectiveness by providing so-called multipliers for each factor (see Section 3.3.5). The answers of one expert had to be removed because of invalid answers caused by a misunderstanding of the questionnaire.

Table 2 Identified relevant influencing factors for defect content

Category : Defect Content - Product		W=0.3778 *
Imp.	**Factor Name**	**Mean** **
1	Novelty to developer	4.143
2	Number and complexity of external interface	4.143
3	Autonomous	4.286
4	Required failure tolerance	5.429
5	Requirement's assumption	5.429
6	Number of component decompositions	6.142
7	Time criticality	6.286
8	Hardware architecture	6.571
9	Role of functionality	6.714
10	Sub-architecture	7.571
11	Legacy part	9.714
12	Memory size	11.429
Category : Defect Content - Project		**W=0.3143**
1	Number of stakeholders/user organizations	1.429
2	Involvement of customer in development	2.714
3	Developer's stress	2.857
4	Size of developer's project team	3.000
Category : Defect Content - Process & Personnel		**W=0.4531** *
1	Project complexity	1.429
2	Schedule adherence	2.174
3	Management of developer	3.000
4	Developer's requirements analysis	3.571
5	Developer's knowledge about tools	4.286

* significant at α=0.05, ** of the rankings provided by the experts

4.3.3 Results and Interpretation: Applicability of the Method (RQ1)

Objective 1 Tables 2 and 3 include Kendall's coefficient of concordance W for each category. We got significant agreement for half of the categories. We observed the highest disagreement between experts for product-related factors influencing effectiveness (W= 0.123) and the highest agreement for process- and personnel-related factors influencing effectiveness (W=0.475).

Objective 2 Finally, we were able to build an initial causal model based on the defined factor selection criterion (following the rigorous-ranking-based approach). The resulting IV&V DCE causal model includes five defect content factors and five IV&V effectiveness factors (see Fig. 7).

Table 3 Identified relevant influencing factors for QA effectiveness

Category : QA Effectiveness - Product		W=0.1230
Imp.	**Factor Name**	**Mean****
1	Consistent terminology of documentation	2.714
2	Documentation completeness	2.857
3	Type of language	3.386
4	Documentation of exceptional behavior	3.714
5	Documentation structure	4.143
6	Figures/charts in documentation	4.286
Category : QA Effectiveness - Project		**W=0.2257**
1	Relationship between developer and IV&V	2.714
2	Management of document change	2.714
3	Involvement of customer in development	3.429
4	Disclosure of electronic file	3.429
5	Experience of IV&V manager	3.571
6	Transparency to stakeholder	5.143
Category : Effectiveness - Process & Personnel		**W=0.4752***
1	IV&V's knowledge of system/operation	1.429
2	Support from developer (supplier)	2.517
3	Supplier performs FTA	4.429
4	IV&V team relationship	4.714
5	Size of the IV&V team for review after RA	5.000
6	Size of the IV&V team for risk analysis (RA)	5.714
7	Supplier performs FMEA	5.857
8	IV&V's experience with tool	6.286

* significant at α=0.05, ** of the rankings provided by the experts

Objective 3 COQUALMO (Huang and Boehm 2005) consists of the Defect Introduction (DI) model and the Defect Removal (DR) model. QA techniques and their intensity are modeled in the DR model. In this study, we analyze IV&V's document review of the requirement specification, which is one driver/factor of the DR model. Therefore, the results of the DR model should be stable and the defect remove effectiveness should be equal for all projects. The DI model defines 21 defect introduction drivers/factors, which are categorized into four groups, Platform, Product, Personnel, and Product. Because the IV&V phase is the requirements analysis phase and IV&V personnel has no detailed information about development project, only five defect introduction factors were measurable. Only two out of five factors correspond to the defect contents factors identified together with the local domain experts (see Table 4).

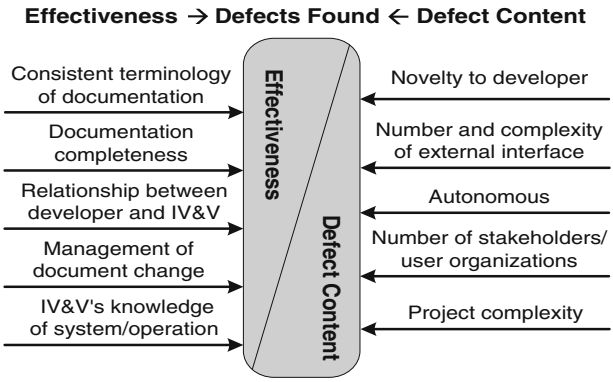


Fig. 7 Resulting DCE causal model

Interpretation The domain experts identified factors other than COQUALMO factors as the most influencing factors. Therefore, it seems reasonable to build context-specific models and not simply reuse existing models with context-independent factors.

4.3.4 Results and Interpretation: Effort for Model Building (RQ2)

The identification of relevant factors was based on the discussion of an existing set of factors between one domain expert and one measurement expert. Relevant factors were selected, categorized, and the best and worst cases were quoted. Thus, additional experts were only involved in answering the two questionnaires for factor ranking and quantification. Figure 8: The median effort for the ranking questionnaire (Quest. 1) was 1 h, and varied between 30 min and 90 min. To answer the quantification questionnaire (Quest. 2), the median effort was 40 min and varied between 30 min and 60 min. On average, the total effort per expert was 112 min; this includes time required for requests.

Interpretation The required level of expert involvement is an important factor for the applicability of approaches that use expert judgment, since especially the most experienced experts are typically limited in their availability (Trendowicz et al. 2008). The required effort per expert (less than 2 h on average) for building the HDCE model was manageable, especially since the DCE causal model has to be built once and can then be applied for all upcoming projects in the defined context. However, considering the total effort for domain experts, it is important to note that the total effort presented does not include the effort for the brainstorming session usually conducted with several domain experts to indentify relevant influencing factors.

Table 4 Comparison: COQUALMO factors and most important factors identified by local expert

	Defect introduction driver	Relation	DCE model (DC factor)	Category
DI1	Required reusability	≠	Legacy part	Platform
DI2	Documentation match to life-cycle needs	~	Documentation completeness	Platform
DI3	Main storage constraint	≠	Requirement volatility	Product
DI4	Required development schedule	≠	Schedule adherence	Project
DI5	Precedentedness	~	Novelty (for Developer)	Project

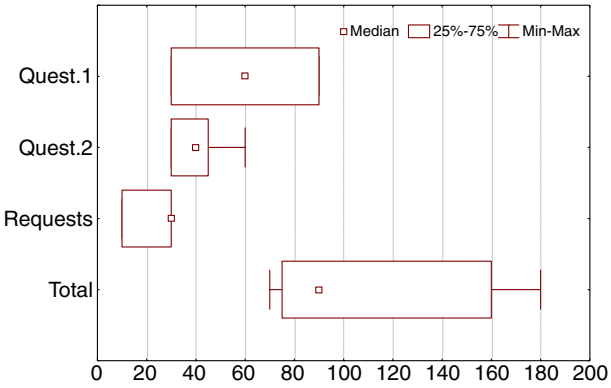


Fig. 8 Box & Whisker Plot of effort of involved experts in minutes

4.4 Model Application: Evaluation of RQ3 & RQ4

After the quantitative causal model had been built in the case study, available legacy data (and, to some extent, expert opinions) were used to evaluate the usefulness of the model for planning and controlling QA activities.

4.4.1 Study Objectives and Design

In order to evaluate the usefulness of the model with respect to *planning QA* (RQ3), it would be most straightforward to compare the predicted quality risk (i.e., remaining defect density) for each project with the actual remaining density (i.e., the actual number of remaining defects per size unit). However, since for the historical projects, no data was available about the number of defects that slipped through the QA activities in SRA, the usefulness of the model with respect to *planning QA* (RQ3) by providing quality risk analysis could not be evaluated directly. Therefore, the plausibility of the model results was evaluated by checking the following two hypotheses:

H3.1: The experts agree on the quality risk assessment results provided by the model for the historical projects.

H3.2: If the model predicts a higher relative effectiveness for a project, more defects are found per size unit, i.e., if a project has a higher relative effectiveness (*EIF*) compared to a second project and the second project has no higher relative defect density (*DDIF*), then the *number of defects found per artifact size unit (DF/Size)* in the first project is higher than in the second project. The same holds for defect density; higher defect density results in more defects found per size unit. Figure 9 illustrates these statements.

The usefulness of the model for *controlling QA* (RQ4) by providing an indicator or thresholds for the number of defects expected to be found by the QA activity is checked by determining the prediction error of the HDCE model, comparing the prediction error with the prediction error of applicable data-based prediction models, and checking the usefulness of each component of the model (defect content factors, effectiveness factors, size information).

H4.1: The *estimation accuracy* of the DCE model is higher than the accuracy of a model using only available measurement data (i.e., not including expert judgment).

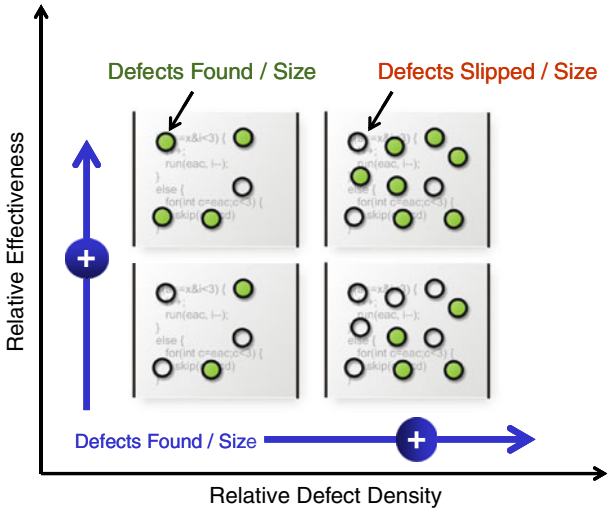


Fig. 9 Higher defect density or effectiveness results in more defects found per size unit

H4.2: Each component of the estimation model (i.e., the artifact size and the expert-based causal model with *EIF* and *DDIF*) contributes to the *estimation accuracy* of the HDCE model.

4.4.2 Study Instrumentation and Execution

To apply the model for planning and controlling QA and check our hypotheses, the used constructs had to be operationalized, data had to be collected, and evaluation procedures needed to be defined (e.g., by selecting appropriate statistical tests).

Relevant Constructs

Number of defects found (DF) is measured as the number of issues that had been found by the IV&V supplier in SRA. The total number of issues is counted, even if certain issues are unrelated to problems of the development project.
Size of artifact (Size) is measured as the number of pages of the document checked in SRA by the IV&V supplier.
Project characterization with respect to the influencing factors is provided on the four-level scales defined by the causal model for each factor (Section 3.3.4).
Estimation accuracy refers to the nearness of an estimate to the true value. In order to remain comparable to other estimation studies, we used common estimation error measures and accuracy measures (Conte et al. 1986), such as *relative error* (*RE*) and *mean magnitude of relative error* (*MMRE*). In the following, we consider a model to be more accurate than another one if its *MMRE* value is lower.

Data Collection The number of *defects found* and the *artifact size* were collected from IV&V activities in SRA for five available historical projects (A to E) by checking the issue reports and SRA documents. The *project characterization* was conducted by one expert who knew the five projects. The result was later discussed with a second expert in order to check the validity of the characterization. Figure 10 provides an overview of the

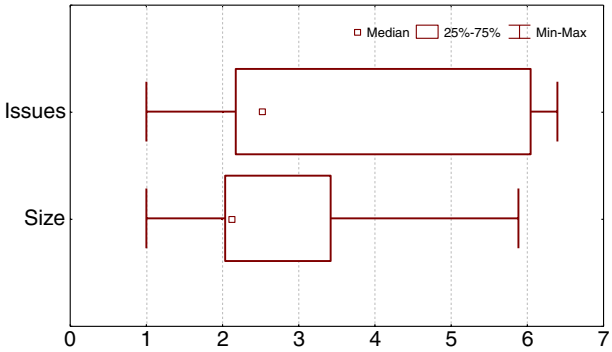


Fig. 10 Box & Whisker Plot of normalized issue and size data from the historical projects

distribution of defects found and artifact size in the historical projects. The values are normalized due to confidentiality issues.

Procedure for QA Planning (RQ3) Based on the project characterization provided by the experts, the resulting distributions for *DDIF* and *EIF* were determined by Monte Carlo simulation for each historical project using the freely available CoBRIX tool (IESE Fraunhofer 2008). Using the *DDIF* and *EIF* distributions, a quality risk chart was created and experts were asked to provide their retrospective opinion about the relative quality risk in the considered historical projects. In a second step, all relationships that could be extracted from the quality risk chart with respect to the assumptions stated in H3.2 were collected in a table. The relationships have the form 'Project A is assumed to have a lower number of defects found per size unit than project B' or '$DF/Size_A < DF/Size_B$' for short. These predicted relationships are then compared with the actual relationships extracted from the historical project data. The absolute and relative number of correctly predicted relationships is calculated. The statistical significance of the result is checked with a one-sided binomial sign test for single samples (Sheskin 2007). As significance level for the test, we chose .05.

Procedure for QA Controlling (RQ4) To evaluate the estimation accuracy, we used *leave-one-out cross-validation* (Allen 1974), justified by the low number of projects available for prediction model building and validation. In our case, this meant that four of the five projects were used to build a prediction model that was applied to estimate the defects found in the fifth 'unknown' project. This procedure was repeated for each of the five projects and the resulting MMRE was calculated. Since Kitchenham et al. (2001) criticized MMRE as a measure of model accuracy, but MMRE is the de-facto standard, we decided to present, in addition to MMRE, box plots of RE values for visual examination of the results as recommended by Kitchenham et al. (2001).

The model was compared to the two reasonable prediction models (H4.1) that can be built with the available project data, namely defects found and artifact size. (1) The DF_{only} *model* assumes an equal number of defects found for each project and takes the median of the defects found in the historical projects to predict the number for the actual one. (2) The *DF+Size model* assumes stable defect density and effectiveness for all projects. Therefore, it uses the median defect density of the historical projects to estimate the defect density of the actual project and then multiplies the estimated defect density with the artifact size of the actual project to predict the number of defects found.

To evaluate the contribution of the components of the model (H4.2)—the expert-based causal model with defect content factors (*DDIF*) and effectiveness factors (*EIF*), and the

artifact size data—we evaluated the accuracy of the model when a certain component was not considered. This was done by setting *DDIF*=0 (*w/o DDIF*), *EIF*=0 (*w/o EIF*), or *size*=1 (*w/o size*) for all projects, respectively.

The observed differences in the magnitude of relative error (*MRE*) was tested for statistical significance by using a two-sided *Wilcoxon test* (i.e., the non-parametric equivalent of the paired t-test, which was not applicable since we could not assume a normal distribution). As significance level, we chose .05.

4.4.3 Result & Interpretation: QA Planning (RQ3)

The quality risk analysis diagram resulting from the model application is presented in Fig. 11. The five dots represent the five historical projects A to E. The *DDIF* and *EIF* values of the project, which represent the mean values of the respective distribution, are used to calculate the relative defect density and relative effectiveness. Due to confidentiality issues, the relative effectiveness and defect density values of all projects are scaled by a fixed factor (f). This means, for example, for project A:

$$relative\ defect\ density = \left(DDIF_A - DDIF_{average}\right) \cdot f$$

$$relative\ effectiveness = \left(EIF_A - EIF_{average}\right) \cdot f$$

In order to check hypothesis H3.1, the experts were asked for their retrospective opinion about the risk of the five projects and agreed on the results that project C was the most risky one and the risk in project A (although many defects were found) was not assessed to be high, especially because of effective IV&V. Projects B and E were considered in the routine catch of IV&V experience in the organization. These results support the plausibility of the model results based on retrospective expert opinion (H3.1).

In order to check hypothesis H3.2, we visually identified all '$DF/Size_x < DF/Size_y$' relationships between projects predicted by the risk chart and compared them with the

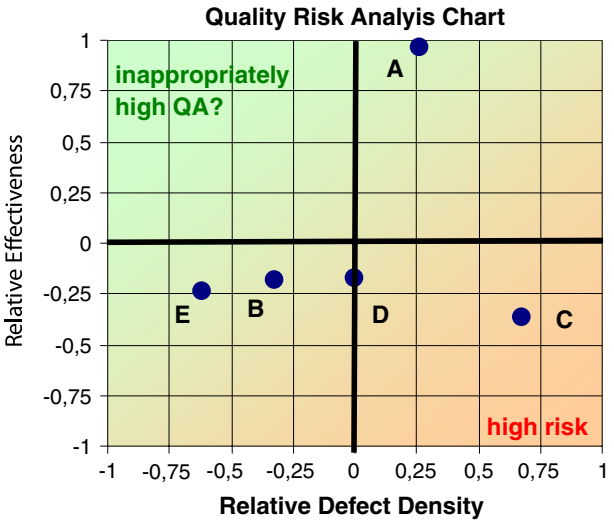


Fig. 11 Historical IV&V projects in risk chart

Table 5 Predicted and actual relationships between defects found per page for different projects

Estimated relationship	Actual relationship	Confirmed
DF/SizeE < *DF/SizeB*	*DF/SizeE* < *DF/SizeB*	Yes
DF/SizeE < *DF/SizeD*	*DF/SizeE* < *DF/SizeD*	Yes
DF/SizeE < *DF/SizeA*	*DF/SizeE* < *DF/SizeA*	Yes
DF/SizeE < *DF/SizeC*	*DF/SizeE* < *DF/SizeC*	Yes
DF/SizeB < *DF/SizeD*	*DF/SizeB* < *DF/SizeD*	Yes
DF/SizeB < *DF/SizeA*	*DF/SizeB* < *DF/SizeA*	Yes
DF/SizeB < *DF/SizeC*	*DF/SizeB* < *DF/SizeC*	Yes
DF/SizeD < *DF/SizeA*	*DF/SizeD* < *DF/SizeA*	Yes
DF/SizeD < *DF/SizeC*	*DF/SizeD* < *DF/SizeC*	Yes

actual '$DF/Size_x < DF/Size_y$' relationships (see Table 5). The *DF/Size* values for the projects are not presented because of confidentiality issues. All nine identified relationships are confirmed by available *DF* and *size* data. For the three projects located on nearly the same effectiveness level (E, B, and D) with increasing relative defect density, the number of defects found per page increases. Project A with high relative effectiveness compared to the remaining projects and project C with high relative defect density both have a higher *DF/Size* value compared to projects E, B, and D. Also, the binomial sign test confirms statistical significance at a p-level < 0.002. This supports the plausibility of the model results based on available historical data (H3.2).

Interpretation Due to the fact that defect slippage data are unavailable, it is not possible to show a direct relation between predicted quality risk and remaining defect density. However, available measurement data as well as the experts' agreement on the model's results for the historical projects indicate the plausibility and the usefulness of model results.

4.4.4 Result & Interpretation: QA Controlling (RQ4)

The accuracy achieved by the HDCE model is measured by an *MMRE* of 29.6%.

Interpretation This value seems good for an initial model when compared to the results of cost estimation models built by applying CoBRA (Trendowicz et al. 2006) (initial model 107%, first iteration 32%), but improvable (fourth iteration 14%). We performed no iterations to improve the model's accuracy due to time constraints and the fear of overfitting the model with respect to the limited number of historical projects.

Table 6 summarize the RQ4 (estimation accuracy) related results. It presents the MMRE for each model, the improvement of the HDCE model as absolute and relative accuracy improvement (measured as MMRE reduction), the p-value of the statistical test, and whether the test results are significant at the chosen significance level.

Reduced estimation error can also be observed visually by checking the box plots in Fig. 12. In addition, Fig. 13 presents the mean relative error and the .90 confidence interval. The accuracy of the confidence interval is questionable due to the limited number of data points. However, the diagram shows that the mean estimation error of the HDCE model is close to zero and has the smallest variance when compared to the results of other models.

Table 6 Overview of prediction accuracy improvement determined by cross-validation

Overview RQ4	HDCE	Hypothesis 4.1		Hypothesis 4.2			
Model		DF_{only}	DF+ Size	w/o DDIF	w/o EIF	w/o Exp.	w/o Size
MMRE	**0,296**	**1,228**	**0,765**	**0,355**	**0,343**	**0,765**	**0,412**
MMRE Reduction	N/A	0,932	0,469	0,059	0,047	0,469	0,116
p-value (Wilcoxon)	N/A	0,043	0,043	0,893	0,686	0,043	0,686
Relative Improvement	N/A	76%	61%	17%	14%	61%	28%
Significance (α=.05)	N/A	**yes**	**yes**	no	no	**yes**	no

Interpretation H4.1 can be confirmed, since both data-based models (DF_{only} and $DF+Size$) provide estimates with significantly higher estimation errors ($MMRE_{DFonly}$ = 122.8%, $MMRE_{DF+Size}$ = 76.5%), with significance being tested using the Wilcoxon test at a significance level of .05.

With respect to H4.2, one can see that a model not considering the factors influencing effectiveness would result in reduced accuracy ($MMRE_{w/oEIF}$ = 34.2%), as would a model not considering factors influencing defect content ($MMRE_{w/oDDIF}$ = 35.4%) or artifact size, i.e., using only expert opinion ($MMRE_{w/oSize}$ = 41.2%). These results show that not only the expert-based components, but also the measurement component contributes to model accuracy. However, the significance of the improvement provided by the different parts of the model could not be shown due to the reduced effect size. Therefore, H4.2 can be confirmed only partially.

4.5 Threats to Validity

The results of any empirical study have to be discussed with respect to their validity. In this section, threats to validity are presented that were identified during and after the conduction of this study and which are considered to be relevant. In our effort not to overlook any major threat, we used a checklist of typical threats (Wohlin et al. 2000). Nevertheless, we

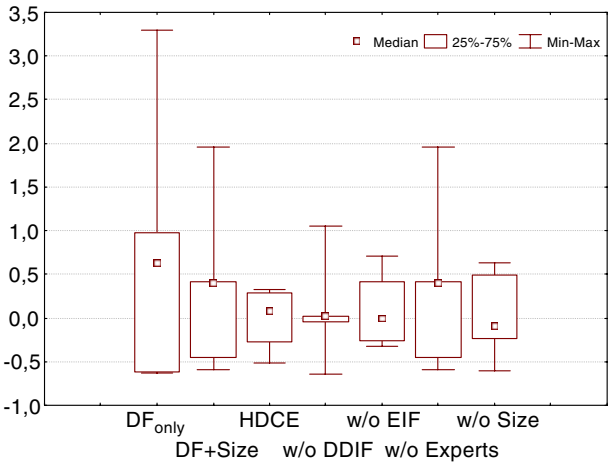


Fig. 12 Box and Whisker plot of relative prediction error (*RE*) determined by cross-validation

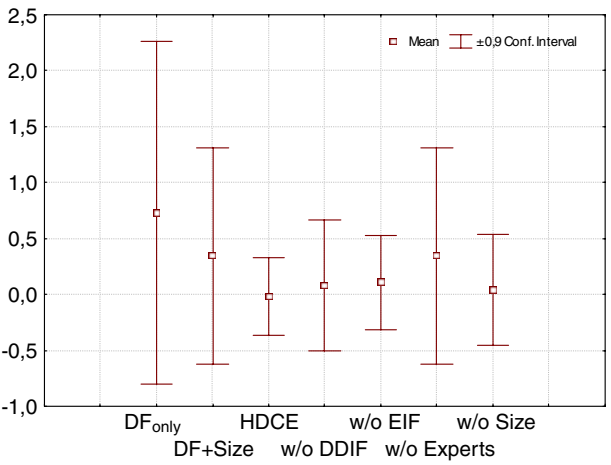


Fig. 13 Whisker plot with mean and 0.9 confidence interval of relative prediction error (*RE*)

have to mention that there may be further threats not identified. In the following, the threats are presented classified into conclusion, internal, construct, and external ones, as proposed by Cook and Campbell (1979).

Conclusion validity deals with the ability to draw (statistically) correct conclusions from the analyzed data. We used only robust non-parametric tests to check our hypotheses, so we see no relevant threat to violating the assumptions of the applied tests. However, only a small number of historical data points (five projects) was available for building and evaluating the HDCE model. Therefore, the statistical power of the applied tests was limited, and we could not show a statistically significant positive impact for all model components (H4.2). As a result, although the numbers suggest a positive impact, due to low statistical power, we can only see a trend suggesting such an impact for all components. A second threat to the conclusion validity can be seen in the measure we used to operationalize the concept defects found, namely, we used the number of issues reported in the historical inspection protocols. This measure intrinsically depends on individual inspectors' understanding of issues and criteria for reporting these issues. Therefore, the measure is, at least to a certain degree, subjective, which reduces its reliability. Yet, all data used in the study were provided by a specific group of experts in one company. Hence, we assume a sufficient degree of homogeneity in the understanding and reporting of issues, which would compensate any introduced bias to a certain extent.

Internal validity can be affected, among other things, by mortality effects, i.e., subjects who dropped out of the study. During this study, the questionnaire for determining the impact of influencing factors was filled out by one of the experts in such a way that the answers could not be used for model building. The reason for the invalid answers was a misunderstanding of the questionnaire by the expert. Since six of seven experts provided valid answers, which were used afterwards for model building, we consider the impact of the missing answers of the seventh expert as limited with respect to the study results. Besides, we did not perform a workshop for introducing the questionnaires to the experts; therefore, there might be some bias of understanding of the questionnaires (instrumentation) and, accordingly, also in the answers provided, which we cannot assess.

External validity considers the generalizability of the results beyond the context of the specific study. One the one hand, the presented study was conducted in a realistic

environment (i.e., in a company with professionals as subjects, and data taken from real projects). On the other hand, the causal model is based only on the answers provided by IV&V personnel from one organization working in the space domain. If we consider another organization in the space domain or even another domain, the resulting model would be different, since the model is specific for the context in which it is built. Nevertheless, the identified factors may be a good starting point for companies working in the space domain that would like to identify factors relevant in their context.

5 Summary and Conclusions

In this paper, we pointed out the lack of empirically validated methods for building context-specific prediction models for planning and controlling quality assurance activities during early phases of the software lifecycle when only limited historical data points and measurement data are available. We also pointed out that the defect content of the investigated artifact and the effectiveness of quality assurance activities are context-specific and affected by several influencing factors.

Therefore, we proposed a hybrid defect content and effectiveness (HDCE) prediction method that allows building context-specific models that consider the most relevant influencing factors in the context. The HDCE method combines available historical project data and expert judgment encapsulated in a reusable quantitative causal model for factors influencing defect content and effectiveness.

The *applicability of the method* (RQ1) was shown by conducting a case study in the context of IV&V for the software requirements analysis phase of critical software in the space development domain. To build the required causal model, the knowledge of seven domain experts was elicited basically with the help of questionnaires, which required, on average, 112 min per expert in total (RQ2).

The usefulness of the resulting causal model was evaluated using measurement data collected for five historical projects (artifact size, number of defects found). The evaluation results suggest the usefulness of the model for *QA planning* (RQ3) by identifying projects with high quality risk. Moreover, this also holds for *QA controlling* by providing (statistically) significant better estimates for the number of defects expected to be found than models using only measurement data (RQ4).

In addition, from the perspective of the IV&V experts, the mapping of the probability distribution for the five historical projects showed the predicted risk of the IV&V strategy and confirmed the experts' impression regarding historically evaluated projects. Consequently, the proposed hybrid prediction model will be integrated into the IV&V planning and monitoring procedures of JAMSS to support QA activities during the software requirements phase.

As a future direction, the HDCE prediction model should be expanded by collecting and including defect data from testing phases to determine defect slippage during earlier phases. This expansion would serve to make the model more precise and would allow predicting absolute defect content and effectiveness values. Furthermore, we plan to expand our model to predict risk exposure during the operational phase (after release), including severity of defects.

Acknowledgment We would like to thank the development project staff and the IV&V staff from the JAXA Engineering Digital Innovation Center (JEDI) at the Japanese Aerospace Exploration Agency (JAXA), where we conducted the case study to construct the hybrid prediction model. We would like to thank the staff of JAMSS, who greatly contributed by answering the questionnaires and giving us historical experience data.

Finally, we would like to thank Adam Trendowicz and Marcus Ciolkowski from Fraunhofer IESE for the initial review of the paper, Sonnhild Namingha for proofreading, and the anonymous reviewers of the International Symposium on Software Reliability Engineering and the Journal of Empirical Software Engineering for their valuable feedback. Parts of this work have been funded by the BMBF SE2006 project TestBalance (grant 01 IS F08 D).

Appendix

Table A1 Major Categories of defect injection and detection factors collected by Jacobs et al. (2007)

Defect content/defect injection factors		Effectiveness/defect detection factors	
1	Developer capability	1	Testability
2	Domain knowledge	2	Product complexity
3	Team composition	3	Quality of documentation
4	Team distribution	4	Change control
5	Collaboration	5	Test planning
6	Business management maturity	6	Management attitude
7	Product complexity	7	Adherence to plan
8	Communication	8	Test process maturity
9	Project management maturity	9	Development process maturity
10	External disturbances	10	Test environment
11	Process maturity	11	Support for testing
12	Change control	12	Product integration
13	Quality of documentation	13	Test capability
14	Requirements	14	Test team cohesion
15	Development environment	15	Team distribution
16	Innovation	16	Test team organization
		17	Communication

Table A2 Defect introduction drivers used in COQUALMO model (Huang and Boehm 2005)

Category	Defect introduction drivers
Platform	Required software reliability
	Database size
	Required reusability
	Documentation match to life-cycle needs
	Product complexity
Product	Execution time constraint
	Main storage constraint
	Platform volatility
Personnel	Analysis capability
	Programmer capability
	Applications experience
	Platform experience
	Language and tool experience
	Personnel continuity
Project	Use of software tool

Category	Defect introduction drivers
	Multisite development
	Required development schedule
	Precedentedness
	Architecture/risk resolution
	Team cohesion
	Process maturity

References

Allen DM (1974) The relationship between variable selection and data augmentation and a method for prediction. Technometrics 16(1):125–127

Aurum A, Petersson H, Wohlin C (2002) State-of-the-art: software inspections after 25 years. Softw Test Verif Reliab 12(3):131–154

Bibi S, Tsoumakas G, Stamelos I, Vlahvas I (2006) Software defect prediction using regression via classification. Int Conf Comput Syst Appl, pp 330–336

Briand L, Freimunt B (2004) Using multiple adaptive regression splines to support decision making in code inspections. J Syst Softw

Briand L, El Emam K, Freimut B, Laitenberger O (1997) Quantitative evaluation of capture-recapture models to control software inspections. 8th Int Symp Softw Reliability Eng, pp 234–244

Briand L, El Emam K, and Bomarius F (1998) COBRA: a hybrid method for software cost estimation, benchmarking, and risk assessment. ISERN-97-24

Briand L, El Emam K, Freimut B, Laitenberger O (2000a) A comprehensive evaluation of capture-recapture models for estimating software defect content. IEEE Trans Softw Eng 26(6):518–540

Briand L, Wüst J, Daly JW, Porter V (2000b) Exploring the relationships between design measures and software quality in object-oriented systems. J Syst Softw 51:245–273

Conte SD, Dunsmore HE, Shen VY (1986) Software engineering metrics and models. Benjamin-Cummings, Menlo Park, CA

Cook TD, Campbell DT (1979) Quasi-experimentation: design and analysis issues for field settings. Mifflin, Boston

Eick SG, Loader CR, Long MD, Votta LG, Wiel SV (1992) Estimating software fault content before coding. 14th Int Conf Softw Eng, pp 59–65

El Emam K, Laitenberger O, Harbich T (2000) The application of subjective estimates of effectiveness to controlling software inspections. J Syst Softw USA 54(2):119–136

Endres A, Rombach D (2003) A handbook of software and systems engineering. Addison Wesley

Fenton N, Neil M (1999) A critique of software defect prediction models. IEEE Trans Softw Eng 25(5):676–689

Fishman GS (1995) Monte Carlo: concepts, algorithms, and applications. Springer Verlag, New York

Freimut B (2006) MAGIC A hybrid modeling approach for optimizing inspection cost-effectiveness. Fraunhofer-IRBVerlag, Stuttgart

Friedman J (1991) Multivariate adaptive regression splines. Ann Stat 19:1–141

Halstead MH (1977) Elements of software science. Elsevier, New York

Huang L, Boehm B (2005) Determining how much software assurance is enough? A value-based approach. In: International Symposium on Empirical Software Engineering, Noosa Heads, Qld., Australia, 17–18 Nov

IEEE (2005) Std. 1012-2004. IEEE standard for software verification and validation. IEEE Comput Soc

IESE Fraunhofer (2008) CoBRIX Tool. http://www.cobrix.org/cobrix/index.html. Accessed 1 May 2008

Jacobs J, van Moll J, Kusters R, Trienekens J, Brombacher A (2007) Identification of factors that influence defect injection and detection in development of software intensive products. Inf Softw Technol 49 (7):774–789

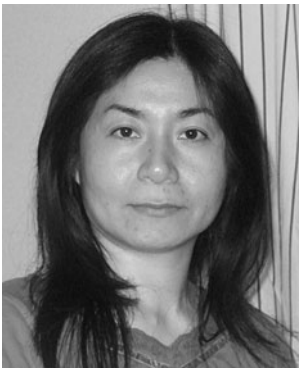
Jones C (1996) Applied software measurement: assuring productivity and quality, 2nd edn. McGraw-Hill, New York

Juristo N, Moreno AM, Vegas S (2002) A survey on testing technique empirical studies: how limited is our knowledge? 1st Int Symp Empir Softw Eng, pp 161–172

Kan SH (2003) Metrics and models in software quality engineering, 2nd edn. Addison-Wesley, Boston

Kendall MG, Smith B (1939) The problem of m rankings. Ann Math Stat 3:275–287
Kitchenham BA, Pickard LM, MacDonell SG, Shepperd MJ (2001) What accuracy statistics really measure. IEEE Softw 148(3):81–85
Kläs M, Trendowicz A, Wickenkamp A, Münch J, Kikuchi N, Ishigai Y (2008) The use of simulation techniques for hybrid software cost estimation and risk analysis. In: Advances in computers, (74)115–174, Elsevier
Kohtake N, Katoh A, Ishihama N, Miyamoto Y, Kawasaki T, Katahira M (2008) Software independent verification and validation for spacecraft at JAXA. IEEE Aerosp Conf
McCabe TJ (1976) A complexity measure. IEEE Trans Softw Eng 2(4):308–320
McKay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21(2):239–245
Meyer MA, Booker JM (2001) Eliciting and analyzing expert judgment. A practical guide. [First publ. by Acad. Press Ltd, London, 1991]. Philadelphia, Pa: Society for Industrial and Applied Mathematics and American Statistical Association (ASA-SIAM series on statistics and applied probability, 7)
Nagappan N, Ball T, Zeller A (2006) Mining metrics to predict component failures. 28th Int Conf Softw Eng, pp 452–461
Nakao H, Yoshikawa S, Port D, Miyamoto Y, Katahira M (2007) Comparing model generated with expert generated IV\&V activity plans. Proc 1st Int Symp Emp Softw Eng Meas: IEEE Comp Soc, pp 71–80
NIST (2002) Planning Report 02-3, The economic impacts of inadequate infrastructure for software quality
Petersson H, Thelin T, Runeson P, Wohlin C (2004) Capture-recapture in software inspections after 10 years research. Theory, evaluation and application. J Syst Softw 72(2):249–264
Ruhe M, Jeffery R, Wieczorek I (2003) Cost estimation for web applications. 25th Int Conf Softw Eng, pp 285–294
Sheskin DJ (2007) Handbook of parametric and nonparametric statistical procedures, 4th edn. Chapman & Hall/CRC, Boca Raton, Fla
Shull F, Basili V, Boehm B, Brown AW, Costa A, Lindvall M, Port D, Rus I, Tesoriero R, Zelkowitz M (2002) What we have learned about fighting defects. 8th Int Symp Softw Metr USA, pp 249–258
Trendowicz A, Heidrich J, Münch J, Ishigai Y, Yokoyama K, Kikuchi N (2006) Development of a hybrid cost estimation model in an iterative manner. 28th Int Conf Softw Eng, pp 331–340
Trendowicz A, Münch J, Jeffery R (2008) State of the practice in software effort estimation: a survey and literature review. Proceedings to the 3rd IFIP TC2 Central and East European Conference on Software Engineering Techniques, Brno, 13–15 October 2008. To appear in Springer LNCS, Springer Verlag, 2009
Vose D (1996) Quantitative risk analysis. a guide to Monte Carlo simulation modeling. Wiley, Chichester
Weller EF (1994) Using metrics to manage software projects. IEEE Comput J USA 27(9):27–33
Wohlin C, Runeson P (1998) Defect content estimations from review data. 20th Int Conf Softw Eng, pp 400–409
Wohlin C, Runeson P, Host M, Ohlsson MC, Regnell B, Wesslen A (2000) Experimentation in software engineering an introduction. Kluwer, Boston, MA

Michael Kläs is a researcher in the Department of Processes and Measurement at the Fraunhofer Institute for Experimental Software Engineering (IESE) in Kaiserslautern, Germany. He received his diploma in computer science (master) from the University of Kaiserslautern, Germany. His research interests include software quality measurement, defect prediction, empirical software engineering, and software cost estimation.



Haruka Nakao is an engineer in the Safety and Product Assurance Department at the Japan Manned Space Systems Corporation. She received her diploma in aerospace engineering (master) from Nihon University, Japan. Her research interests include software safety, model-based software engineering, and software quality measurement. She was a guest researcher at the Fraunhofer Institute for Experimental Software Engineering in 2007. She is a member of IEEE.

Frank Elberzhager is a researcher in the Department of Testing and Inspections at the Fraunhofer Institute for Experimental Software Engineering (IESE) in Kaiserslautern, Germany. He received his diploma in computer science (master) from the University of Kaiserslautern, Germany. His research interests include software quality assurance in general and static quality assurance techniques such as software inspections or reviews in particular.

Jürgen Münch is Division Manager for Quality Management at the Fraunhofer Institute for Experimental Software Engineering (IESE) in Kaiserslautern, Germany. Before that, he was Department Head for Processes and Measurement at IESE and an executive board member of the temporary research institute SFB 501, which focused on software product lines. Jürgen Münch received his Ph.D. in computer science from the University of Kaiserslautern, Germany. His research interests in software and systems engineering include: (1) modeling, measurement, and evolution of software processes and resulting products, (2) software quality assurance and control, (3) technology evaluation through experimental means and simulation, (4) software product lines, (5) technology transfer methods. He is a member of ACM, IEEE, the IEEE Computer Society, and the German Computer Society (GI), as well as a member of the program committees of various software engineering conferences. He has been co-organizer or program co-chair of several renowned software engineering conferences such as ESEM. He is a co-recipient of the IFIP TC2 Manfred Paul Award for Excellence in Software Theory and Practice.