

# Using Inspection Data for Defect Estimation

*To control projects, managers need accurate and timely feedback on the quality of the software product being developed. I propose subjective team estimation models calculated from individual estimates and investigate the accuracy of defect estimation models based on inspection data.*

**Stefan Biffl**, Vienna University of Technology

**P**roduct quality directly relates to project cost and schedule estimation; for example, undetected defects in a key work product—such as a requirements document—might lead to time-consuming adjustments. Thus, from the early project stages on, developers and project managers need to update their estimates of software project schedules and activities with feedback from the development process. A major aspect of this feedback is data on work-product quality levels (number

and type of deviations from specified quality goals).<sup>1,2</sup> Project managers can compare that data to the quality goals in the project plan, which were the basis for the initial estimate. Software product inspection is effective for determining the quality level.

To assess the credibility of the initial cost and schedule estimates, the project manager must know how many defects the software product contains (almost always more than the number of defects found during inspection). One way to assess product quality is to monitor product defects throughout development and operation. However, because of this process's time lag, the information is not available in time for developers to decide on appropriate development and quality assurance activities. Another approach uses estimation models based on the number of defects found after a work product inspection to estimate a product's total number of defects. In his recent software engi-

neering textbook,<sup>3</sup> Watts Humphrey proposes a simple objective defect estimation model (DEM) to control project activities. Empirical evidence on the applicability and accuracy of product quality estimation models based on sound project data can encourage wider use of such models.

In this article, I investigate the accuracy of objective and subjective DEMs based on inspection data. Objective DEMs do not depend on personal opinion but require input from a high-quality data collection process. Subjective estimates, conversely, are relatively cheap and easy to obtain but, by definition, depend on the knowledge and capability of the individual estimator (in this case, an inspector who read the inspection object carefully and reported a list of defects).

I propose models to calculate most likely team defect estimates and a confidence interval from subjective individual defect estimates. I report on data from an experiment

to evaluate the accuracy of subjective team estimates and objective DEMs on the number of defects in a requirements document. Because the number of defects in the inspection object is known, we can objectively assess the accuracy of the DEMs—that is, their usability in providing feedback on product quality for project control.

### Project Control with Product Quality Assessment

A large part of project management is risk management. Therefore, project cost estimation strategies must consider risk assessment to yield realistic results.<sup>1</sup> The quality of work products and the effectiveness of quality assurance methods are important risk factors for project planning and control.

Estimating software project costs and schedules anticipates quality goals for the software product to be delivered. The quality levels of the final software product depend on the respective quality levels of development work products. Key requirements for well-founded estimates are the availability and use of sound data as well as some documented rationale for deriving the estimate. If the project-estimation approach allows developers to define software quality levels for those work products and link these to results of the development process, an assessment of the quality levels of these work products can become a basis to adjust initial cost and schedule estimates.

Cost estimation without credible data is widespread in practice, often for lack of a data basis on properties of past projects.<sup>4</sup> Creating an adequate data basis as required by sophisticated cost estimation models (for example, Cocomo II<sup>5</sup>) becomes even more difficult with changing development paradigms, new development methods, and new computing environments that quickly make historical data obsolete. Feedback on the quality levels of development products is crucial to assess a given estimate, particularly in projects for which no historical data exists to support a current cost and schedule estimate.

### Product Quality Assessment

Along the development life cycle, particularly in the early stages of software development, *inspection* of software documents is an effective quality assurance measure to

detect defects and provide timely feedback on quality to developers and managers. Inspection denotes the verification of software documentation by a team of inspectors<sup>2</sup> with defect-detection, meeting, and repair steps. The defect detection step is an individual activity, implying no interaction among the inspection team.

After inspection, developers and project managers can analyze the retrieved data to evaluate the quality of the work product and development and quality assurance processes. The number of defects found during inspection is not a sufficient criterion for these quality evaluations, because bad products with a bad inspection process might rest undetected while very good products might needlessly be elected for another inspection cycle. An approach to overcome this problem is to estimate the number of defects originally present in the document. With this estimate, the project manager can identify relationships between product quality, inspection process quality, and invested time and costs.

Similar to traditional cost estimation strategies, defect content estimation approaches based on historical project data exist.<sup>6</sup> Project managers must use these approaches with care because such historical data is often unavailable or inapplicable to the project at hand.

In this article, I solve this problem with DEMs that use data from inspection of development work products. These models yield an estimate for the most likely number of defects and a confidence interval to give the project manager information on the defect estimate's probability distribution.

### Objective Defect Estimation Models

The investigation employs capture—recapture models and the detection profile method (DPM).<sup>7,8</sup> CR models, originally developed to estimate the size of closed animal populations,<sup>9</sup> are based on the set of defects found by a team of inspectors. There are four CR models, which differ in their assumptions on the probability of defects to be found and of inspectors to find defects. Each model has at least one estimator (a formula based on the model's assumptions) to calculate the defect estimate. In the experiment, the estimator calculated the defect estimate's most likely value and a 95% con-

**Feedback on the quality levels of development products is crucial to assess a given estimate.**

**Table 1****Capture-Recapture Model**

Model	Model assumptions	Estimators
M0	Defects have equal probability of being detected. Inspectors have equal ability to detect defects.	M0: maximum likelihood. <sup>9</sup>
Mt	Defects have equal probability of being detected. Inspectors have different abilities to detect defects.	Mt: maximum likelihood. <sup>9</sup> Mtc: Chao's estimator. <sup>8</sup>
Mh	Defects have different probabilities of being detected. Inspectors have equal ability to detect defects.	Mh: Jackknife. <sup>9</sup> Mhc: Chao's estimator. <sup>8</sup>
Mth	Defects have different probabilities of being detected. Inspectors have different abilities to detect defects.	Mthc: Chao's estimator. <sup>8</sup>

confidence interval (see Table 1).

Similar to CR models, the DPM builds on the number of inspectors who found a given defect. The DPM sorts the set of defects descending by the number of inspectors who found a given defect. The DPM uses linear regression to fit an exponential function to these data points. The estimate is the point at which the function falls below a given threshold.<sup>7</sup>

According to Lionel Briand, Khaled El Emam, and Bernd Freimut,<sup>7,8</sup> objective DEMs tend to underestimate and yield extreme outliers for special cases. The authors suggested an approach to combine both CR and DPM DEMs, which resulted in improved estimation performance.

### Subjective Defect Estimation Models

Subjective defect estimation strategies ask knowledgeable people to supply a number or set of numbers that describe a software product's set of defects. These estimates can easily be obtained from an inspection team—even from a single inspector. An important advantage of subjective estimation is that it does not require complex data measurement. Others<sup>10,11</sup> have reported encouraging results from individual subjective estimation of defects in code modules.

The basis for a serious estimate should be well-structured information on the product and the set of defects to be estimated. So, successful inspection is an excellent preparation for estimating the number of defects remaining in the inspection object.<sup>2</sup> Khaled El Emam, Oliver Laitenberger, and Thomas Harbich were the first to evaluate an experiment on subjective defect estimation with data from an inspection.<sup>11</sup>

In the work reported here, I start with the independent estimates of individual inspectors who read the requirements documents by themselves. Then, I combine these indi-

vidual estimates with a weighted average model. This has the advantage that such team estimates are less susceptible to extreme outliers, because they can compensate for individual extreme estimates and might favor inspectors with a better estimation basis.

I use an interval estimate to give the inspector the opportunity to express his or her confidence as a range of values, the most likely value, and a minimum and maximum value.<sup>12</sup> If an estimator uses a point estimate, he or she might bias the most likely estimate to be on the safe side regarding a given decision.

The models calculate team estimates from individual three-point estimates and propose three combination models and three kinds of weights.

The *Largest Interval* (LI) model calculates the minimum and maximum team estimate from the most extreme individual values. The most likely value is the average of the extreme team values. This model does not use weights.

The *Weighted Average of Individual Estimates* (WAE) model calculates the weighted average for the team estimate directly from the individual estimates:

$$N_j = \frac{\sum_{k=1}^S (\eta_j^k \cdot \omega^k)}{\sum_{k=1}^S \omega^k} \quad (1)$$

where

- $S$  is the inspection team size (four to six inspectors in this experiment),
- $k$  is the inspector identifier in a team,
- $\eta_j^k$  is the estimate from  $k$  of the total defects present in the document before inspection, where the index  $j$  indicates whether the minimum (*min*), most likely (*ml*), or maximum (*max*) number of defects is estimated.
- $N_j$  is the team estimate of the total defects present in the document before inspection, and
- $\omega^k$  is the weight of  $k$ 's estimate.