A Correlational Study on Four Size Measures as Predictors of Requirements Volatility

Annabella Loconsole
Department of Computing Science
Umea University
SE-90187 Umea Sweden
bella@cs.umu.se

Jurgen Börsler
Department of Computing Science
Umea University
SE-90187 Umea Sweden
jubo@cs.umu.se

ABSTRACT
Requirements volatility is an important risk factor for software projects. Software measures can help in quantifying and predicting this risk. In this paper, we present a correlational study with the goal of predicting requirements volatility for a medium size software project. The study is explorative, i.e. we analyse the data collected for our measures to find out the best predictor. To our knowledge, no empirical validation of requirements change measures as predictors has been performed in an industrial setting. Based on the data collected from two industrial software projects for four measures of size of requirements (number of actors, use cases, words, and lines), we have built and evaluated prediction models for requirements volatility. These models can help project managers to estimate the volatility of requirements and minimize the risks caused by volatile requirements, like schedule and costs overruns. Performing a cross systems validation, the best model showed a MMRE=0.25, which can be considered reliable. Although our models are likely to have only local validity, the general method for constructing the prediction models could be applied in any software development company. In an earlier study, we showed that decisions solely based on developers’ perception of requirements volatility are unreliable. Predictions models, like the one presented here, can therefore help taking more reliable decisions.

Categories and Subject Descriptors
D.2.8 [Software Engineering]: Metrics—Process metrics

General Terms
Experimentation, Measurement

Keywords
Requirement, Prediction Model, Empirical Validation, Correlational Study

1. INTRODUCTION
Requirements change frequently, even during later stages of the development process [28]. Software requirements that change often are usually said to be volatile. High volatility can cause cost and schedule overruns, making the goals of the project hard to achieve. Studies show that requirements volatility has a high impact on project performance [37, 41, 45]. Since we cannot expect requirements to be stable, even when requirements engineering tasks are performed well, we should at least carefully monitor and control them throughout the software life cycle. Monitoring requirements volatility usually involves measuring trends or percentages of changes to requirements, and quantifying and predicting changes to requirements. By anticipating a certain level of volatility, project managers can take appropriate actions in order to decrease project risks. For instance, they can assign extra resources to critical requirements, postponing delivery dates for the part of the software system including critical requirements and re-estimate the overall cost.

In this paper, we describe a correlational study with the goal of empirically validating four measures of size as predictors of requirements volatility. After validating the measures theoretically, we built four prediction models using data collected for a medium-size software project developed at BAE Systems Hägglunds AB, Sweden. We then evaluated the accuracy of one model by applying it on a set of data collected for a second project at the same company. The results show that the size measure “number of lines” (of a requirements document) is a good predictor of volatility. Other size measures (number of actors, number of use cases, number of words) were not found to be significant predictors.

The present work is unique regarding two aspects. First, we aim to predict requirements volatility, while volatility is usually chosen as an independent variable, i.e. as predictor of other software or project attributes, like for example in [1, 19, 24, 41]. Second, we are concerned with volatility of smaller units of requirements, instead of treating volatility as a property of the whole set of requirements of a project. This gives project managers a more fine-grained tool for requirements management.

The remaining part of the paper proceeds as follows: section 2 describes related work regarding empirical validation in general and work related to requirements volatility in particular. Section 3 briefly summarises earlier works by the authors investigating the relationship between measures of
size of requirements and number of changes to requirements. Section 4 describes the goals, hypotheses, theoretical validation of the measures, and data collected in the present correlational study. This section also contains a theoretical validation of the measured used. The construction of the prediction models is described in section 5 and their validation is presented in section 6. Threats to validity are discussed in section 7. Finally, discussions and conclusions are presented in section 8.

2. RELATED WORK

Although requirements volatility is a well-studied area, there is relatively little empirical research. The empirical research available is mostly concerned with the impact of requirements volatility on software or project attributes, like for example project performance or risk [1, 45], software maintenance [41], or defect density [24, 35].

Many measures related to requirements volatility have been proposed in the literature. Measures to assess requirements volatility on software or project attributes, like for example project performance or risk [1, 45], software maintenance [41], or defect density [24, 35].

The goal of the present study is to analyse the ability of four specific measures to predict the volatility of requirements, using two data sets from two different projects (see table 1). Prediction models were constructed applying linear regression analysis using the data set from project A. The best model was then validated using the dataset from project B. We choose linear regression because it is suggested to predict interval and ratio scale dependent variables [6], which is our case. Data collection was semi-automatic, carried out by the authors by studying the documentation of the projects A and B. From the first available revisions of requirements documents, all files were analysed following the rules described in appendix A.

The decision to perform a correlational study was based on the widespread usage of this kind of study in the field. They are often the only option in industrial settings [6].

4. DESCRIPTION OF THE PRESENT CORRELATIONAL STUDY

The Spearman correlation coefficient was calculated between each measure of size of requirements documents and the size of change to requirements documents. Requirements in the project were described in terms of use cases and each requirements document contained one use case model. The size of a requirements document was measured in terms of number of lines, number of words, number of use cases, and number of actors.

The data analysis showed a high correlation between each of the size measures and the total number of changes. This suggests that the measures of size of requirements documents are good indicators of the number of changes for (use case-based) requirements documents.

For the second goal above, we could not find significant correlations between any of our four volatility measures and the rating of volatility by the experts. This implies that the developers’ perceptions of number of changes were not good indicators of requirements volatility for the project analysed.

4.1 Context of the study

We analysed and collected data from the use case-based requirements specifications\(^2\) of two different software projects performed at BAE Systems Hägglunds AB, Sweden (see table 1). The company produces automotive systems with embedded software and is ISO9001 and ISO14001 certified. At the time of the analysis the software systems had been in operation for a number of years, and the projects were described as stable. The data collected was the size of requirements specifications, the number of changes to requirements specifications, and the number of actors involved in the specification creation and maintenance process.

\(^1\)Sometimes, the word stability is used instead of volatility. For instance, a definition of “degree of stability” of requirements is presented in [22]. Both terms are used conjunctly in [20, 38]. In our opinion, the words are antonymous.

\(^2\)We are aware of the fact that some researchers do not consider use cases to be requirements [17, 27, 40, 43].
in operation for approximately 24 months. The Rational Unified Process (RUP) was used in both projects.

The goal for project A was to develop external diagnostics software for personal computers. This software system comprises fourteen small use case models (UCM) with a total of 39 use cases. According to the project terminology, a file containing a UCM is made up of an introduction, a revision table, a use case diagram, and a description of all actors and use cases (see [32] for the UCM template used in project A). Use case modelling was the only technique used in this project for describing functional requirements. The vision document contained only sketchy, high level requirements and was not analysed in detail. Seven non-functional requirements were described in one additional file.

Project B developed an information and control system for the vehicles constructed by the company. This software system comprises 22 use cases and other non-functional requirements described in two separate documents in textual format (non use case-based). According to this project’s terminology, a use case description contains the following sections: overview, revision history, references, description, state-diagram, normal flow, alternative flows, special requirements, start conditions, end conditions, and extension points. The actors of this system were described in a higher level requirements specification document called “use case summary”.

As can be observed from table 1, the documentation for projects A and B did not match completely, even though the projects were developed in the same company. No developer worked on both systems. The objects chosen for the study were the requirements documents of the two projects described above. In project A we analysed fourteen files, each containing a very small use case model. In project B we analysed twenty-two files, each containing one use case. Other documentation, used to understand the projects A and B, were vision documents (the top level requirements specification), the use case summary (where we counted the actors), project plans, iteration plans, and test plans.

The entities analysed in the two projects were requirements documents. Intuitively, the larger the document the more changes there are. Therefore, we believe that the size of requirements is an important factor affecting volatility. The size measures “number of actors interacting with the use cases described in the file” (NUC), “number of lines per file” (NLINE), “number of words per file” (NWORD), and “number of use cases per file” (NUC), are the independent variables chosen for this study. As suggested by Fenton [15], size can be seen as composed of length, functionality, and complexity. In our case, NLINE and NWORD are measures of length, NACTOR is a measure of complexity, and NUC is a measure of functionality. These measures were calculated by the authors using a computerized tool. Although the measures NLINE and NWORD are very similar (see section 5.2), we collected data for both measures to compensate for possible differences caused by formatting and style.

Selecting NLINE as independent variable might seem controversial. Like lines of code (LOC) as a size measure for program size, NLINE depends on the language used and formatting style. As pointed out by [3], what we actually want to measure is how much “knowledge” is described. However, there is no reliable way to measure knowledge in a quantitative way. Another possible choice for the independent variable could have been use case points (UCPs) [2, 40]. However, UCPs are not generally applicable. First of all, it is only applicable to use case-based requirements. The definition of UCPs is furthermore based on a classification of use cases and a number of environmental factors (similar to the cost drivers in COCOMO [5]). This information was not available for our projects. Furthermore, the classification of use cases is a subjective activity, therefore it is not possible to collect UCPs automatically.

Further choices for independent variables could be measures like “number of dependencies between use cases” or “number of steps in scenarios”, which arguably capture the complexity of use cases better than NLINE, NWORD or NACTOR. However, these measures are highly dependent on the use case format used in a particular project. If use cases are described in plain text only (the most basic format), this information might not be available. Therefore, we discarded such measures for the sake of generality.

Volatility depends on many different factors, not only the size of requirements. It might therefore be useful to include project factors (like the type of project or system, type of development process, team size or amount of communication or collaboration between stakeholders) among the indepen-

<table>
<thead>
<tr>
<th>Table 1: Key data of the two projects analysed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project A</strong></td>
</tr>
<tr>
<td>Number of developers</td>
</tr>
<tr>
<td>Project duration</td>
</tr>
<tr>
<td>Use case documents (UCD)</td>
</tr>
<tr>
<td>Average UCD size in kB (Median)</td>
</tr>
<tr>
<td>Total number of req. documents</td>
</tr>
<tr>
<td>Vision/overview documents</td>
</tr>
<tr>
<td>Supplementary req. documents</td>
</tr>
<tr>
<td>Total number of use cases</td>
</tr>
</tbody>
</table>
dent variables. Also factors related to the changes affect volatility, like the type, timing, and urgency of a change. Ideally one would wish to determine volatility using historical data representing all the factors above. In practice, however, only the most relevant variables for which reliable data are available are useful. Further variables may be incorporated at a later stage as understanding grows.

4.3 Dependent variables

To investigate the relationship between requirements size measures and requirements volatility one needs a suitable and practical measure of volatility as the dependent variable of our study. Theoretical definitions of requirements volatility are presented in [4, 36, 38, 39], while operational definitions can be found in [4, 11, 21, 32, 36, 38, 41]. Baumert and McWhinney [4] suggest measuring source and state of change, while Nurmuliani et al. [36] take into consideration the source of change in their theoretical definition of volatility. All other definitions are purely quantitative, i.e. they

1. focus on the amount of changes (additions, deletions, and modifications) to requirements and
2. do not consider the cause of change and the semantics of a change, i.e. in what way a particular change impacts development.

Likewise, we define requirements volatility as the amount of changes to a requirements document over time and measure it as the sum of the change densities of a requirements document.

\[ \text{Volatility} = \sum_{i=1}^{N_{\text{REVISION}}} \frac{N_{\text{CHANGE}}}{N_{\text{WORD}}} \].

Our operational definition of requirements volatility is a function of number of changes (NCHANGE), time measured in number of revisions (NREVISION), and size of the requirements document measured in NWORD. NCHANGE is a count of changed words, therefore, NWORD was chosen to calculate the change density (having the same unit of measurement). NREVISION is a count of the revisions for a file, where a revision is a version of a file with a unique identifier.

There an important difference between our operational definition of volatility and the ones described in the literature. While common definitions consider volatility as a property of all requirements of a project, we look at volatility document by document. This more fine-grained view of volatility makes it possible to distinguish units of requirements that are particularly volatile from those that are more stable.

Please note that we do not perform any cause-effect or impact analysis of individual changes to requirements. Such qualitative analyses would require other types of measures, like for example the type of a change or the number of artefacts affected by a change. Such data is usually not available early on in the development.

Our dependent variable has been determined by counting the changes from one version of a document to the next by means of a tool. When the available change data was considered incomplete\(^3\), we discarded the corresponding files from our analysis. A detailed description of the counting rules can be found in appendix A.

4.4 Theoretical validation

Before describing the collected data and the analysis of it, we must be sure that the measures we are using are valid, i.e. that the measures accurately represent the attributes they are supposed to quantify [26]. There are two different kinds of validations which both should be performed with a successful outcome in order for a measure to be valid: theoretical validation and empirical validation. The goal of theoretical validation is to make sure there is a relationship between the measure and an internal attribute\(^4\). Empirical validation, instead, is done to show that the measure is connected to some external attribute. Empirical validation of the volatility measure proposed in the present paper is described in section 5. The theoretical validation is described below (see table 2 for a summary).

We choose to theoretically validate our measures by applying Briand et al.’s approach [8] because the properties they define for size, complexity, length, coupling, and cohesion match our case well. By applying the properties for size, complexity and length, we find that all our measures satisfy the size properties, while length and complexity properties are not satisfied.

Below, we show the application of the size properties to the measure NWORD as defined in Appendix A. In our context, a system \( S = < E, R > \) is a single requirements document (describing a use case or a use case model). The elements \( E \) of a system are then the words in the document. Since we don’t give any meaning to the words in a document or their relationships, we can consider the set of relationships \( R \) between elements in the system \( S \) as empty.

1. Non-negativity. The size of the system is non-negative, \( \text{size}(S) \geq 0 \). This property is satisfied for any text document.

2. Null value. The size of a system \( S = < E, R > \) is null if \( E \) is empty, \( E=\emptyset \Rightarrow \text{size}(S) = 0 \). This property is also satisfied; if a requirements document is empty, then \( \text{NWORD} = 0 \).

3. Module Additivity. The size of a system \( S = < E, R > \) is equal to the sum of the sizes of two of its modules \( m1 = < E_{m1}, R_{m1} > \) and \( m2 = < E_{m2}, R_{m2} > \) such that any element of \( S \) is an element of either \( m1 \), or \( m2 \). This property is also satisfied considering that we do not assign any meaning to the words in a document and consider all occurrences of words as unique elements. I.e. if we consider a requirements document composed of 2 disjoint sets of words, then the number

\(^3\)We considered change data as complete, if there were at least two revisions for each requirements document analysed.

\(^4\)An internal attribute is a property of a software artefact that can be measured by observing the artefact by its own, separate from its behaviour. An external attribute is a property of a software artefact that can be measured by observing the artefact in its environment [15].
of words of the document is the sum of the number of words of the two modules.

Analogously, we can prove that all other measures (NUC, NACTOR, NLINE, NREVISION, and NCHANGE) are also size measures. However, we cannot show that NACTOR is a complexity measure according to Briand et al.’s definition. According to Briand et al., complexity is a system property that depends on the relationships between the elements. The complexity of a system increases when adding relationships among the elements of a system. In our case, only the elements are counted and not the relationships among the elements. Therefore the complexity properties (in particular property 4) do not apply in our case.

### 4.5 Hypothesis

The hypothesis of the study is the following: the size measures NACTOR, NUC, NWORD, and NLINE are good predictors of requirements volatility. Our hypothesis is built on the idea that larger requirements are affected by changes more than smaller ones, because they contain more information. Although quite intuitive this relationship has not been scientifically proven yet. In particular it would be interesting to know more about the character of this relationship (e.g., linearity).

The relationship between volatility and our four size measures of a requirements document could be causal. However, in this study we only looked at whether a relationship exists, and whether our measures of size can be used as predictors of requirements volatility. The study does not make any claims with respect to causality which can only be proven by performing controlled experiments [42]. In industrial environments it is usually hard to perform controlled experiments, since it is difficult to control variables such as project length, the experience of customers and developers in specifying requirements, the techniques used to collect and document requirements, or the organisation’s maturity. In academic environments, on the other hand, projects are usually to small in size and duration to acquire a sufficient number of data points [31].

### 5. CONSTRUCTION OF PREDICTION MODELS

In our previous studies (see section 3), we found a strong correlation between all four size measures and total number of changes. Based on those results, we have analysed the ability of our measures to predict the volatility of requirements.

The data analysis is obtained by following the procedure suggested by Briand et al. [6, 7] or statistics books such as Draper and Smith [14]. Descriptive statistics based on data sets A and B are described in section 5.1. The following principal component analysis (5.2), univariate regression analysis (5.3), multivariate regression analysis (5.4), sanity tests on the regression models (5.5), and evaluation of goodness-of-fit (5.6) were performed only on data set A. In section 6 we perform cross-validation by applying one of the prediction models on data set B.

The choice of the modelling technique for the regression analysis is mostly driven by the nature of the dependent variable; its distribution, its scale and whether it is discrete or continuous.

#### 5.1 Descriptive statistics

Table 3 shows the following descriptive statistics for data sets A and B; the range from minimum to maximum value (column Range), the total number of elements (Total), mean value (Mean), mean standard error (SE Mean), standard deviation (StdDev), variance, median, and inter quartile range (IQR).

In both data sets, the mean values of NCHANGE and of Volatility are larger than their median, especially in data set B. Furthermore, the mean of NCHANGE in dataset B is higher than the mean in data set A. This implies that the prediction models, built based on data set A, will probably predict low volatility for data set B. NACTOR, NUC and NREVISION have relatively low mean values, standard deviations, and variances in both data sets. However, NACTOR and NUC can be expected to be low, since use case models should be kept small according to common guidelines [23, 30]. The totals of NACTOR in the two data sets are comparable, while the ranges and median values are quite different. This means that prediction models based on NACTOR will probably have low accuracy when applied to data set B. In data set B, NUC has almost zero variance because each file describes only one use case (except one file which described two use cases), while in data set A each file describes a whole use case model. The mean values of NWORD and NLINE in data set B are larger than their median values due to some outliers.

Except for the measure NUC, the ranges of the measures in data set B are larger than in data set A. The totals are in the

---

Table 2: Theoretical validation of all measures

<table>
<thead>
<tr>
<th>Property (c.f. Briand et al. [8])</th>
<th>Satisfied?</th>
</tr>
</thead>
<tbody>
<tr>
<td>General properties for all measures</td>
<td>yes</td>
</tr>
<tr>
<td>1. non-negativity</td>
<td>yes</td>
</tr>
<tr>
<td>2. null value</td>
<td>yes</td>
</tr>
<tr>
<td>For size measures only</td>
<td></td>
</tr>
<tr>
<td>3. module additivity</td>
<td>yes</td>
</tr>
<tr>
<td>For complexity measures only</td>
<td></td>
</tr>
<tr>
<td>3. symmetry</td>
<td>yes</td>
</tr>
<tr>
<td>4. module monotonicity</td>
<td>no</td>
</tr>
<tr>
<td>5. disjoint modules additivity</td>
<td>yes</td>
</tr>
</tbody>
</table>

---

5Extremely high or low measurements will not affect the median as much as they affect the mean. Thus, when we deal with skewed populations we may prefer the median to the mean to express central tendency [44].
order of two times of data set A for the measures NLINE, NWORD, NCHANGE and Volatility. In general, in data set B the measures have larger variation than in data set A. Except for NUC in data set B, our measures have all more than five non-zero data points. There is therefore sufficient variance in all the measures to proceed with the analysis [7].

The subsequent analyses will be done on data set A. After that, the constructed prediction model will be applied on data set B for cross validation.

5.2 Principal component analysis

Principal component (PC) analysis is performed to find the independent variables with high loadings, i.e. those independent variables that best explain the variance in the data. For a set of n measures, there are at most n orthogonal PCs, which are calculated in decreasing order of variance they explain in the data set. Table 4 shows the results of the varimax rotation performed on data set A. This technique allows to identify a clearer pattern of loadings. For each PC, we provide its eigenvalue, the variance of the data set explained by the PC (in percent), and the cumulative variance. Absolute values above 0.7 are set in boldface.

<table>
<thead>
<tr>
<th>Measure</th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>PC4</th>
</tr>
</thead>
<tbody>
<tr>
<td>NACTOR</td>
<td>0.3082</td>
<td>0.6318</td>
<td>0.2543</td>
<td>0.0308</td>
</tr>
<tr>
<td>NUC</td>
<td>0.771</td>
<td>0.158</td>
<td>0.064</td>
<td>0.008</td>
</tr>
<tr>
<td>NLINE</td>
<td>0.771</td>
<td>0.929</td>
<td>0.992</td>
<td>1.000</td>
</tr>
<tr>
<td>NCHANGE</td>
<td>0.270</td>
<td>0.150</td>
<td>-0.951</td>
<td>-0.017</td>
</tr>
<tr>
<td>NREVISION</td>
<td>0.731</td>
<td>0.485</td>
<td>-0.437</td>
<td>-0.198</td>
</tr>
<tr>
<td>NWORD</td>
<td>0.386</td>
<td>0.909</td>
<td>-0.156</td>
<td>-0.016</td>
</tr>
<tr>
<td>Volatility</td>
<td>0.882</td>
<td>0.380</td>
<td>-0.277</td>
<td>0.042</td>
</tr>
</tbody>
</table>

As we can see in table 4, NLINE and NWORD get high factor loadings in PC1. They express the same orthogonal dimension, i.e. the size of requirements documents. NUC and NACTOR get very high loadings in PC2 and PC3, respectively. PC2 expresses functionality, PC3 complexity, and PC4 expresses other factors. Among the four PCs in table 4, two of them show sufficient variance, as can be seen in the scree plot in figure 1. The two PCs with high loadings capture about 93% of the variance in the data set. PC3 and PC4 account for too small percentages of variance and will therefore be excluded from further analyses, i.e. the measure NACTOR will not be considered in the multivariate analysis.

5.3 Univariate regression analysis

The results of the univariate analysis on data set A are shown in table 5. Regression analysis is conducted here to investigate the importance of each of the four size measures (the independent variables) in determining volatility (dependent variable). We test the hypothesis that the independent variables are significantly correlated with the dependent variable. This is also a way to screen out measures that are not likely to be significant predictors in multivariate models. We applied linear regression to each of the four measures (also called ordinary least squares regression), which is most suitable to predict a dependent variable at the interval or ratio scale [6]. Two of the four measures had a statistically significant positive relationship with the dependent variable Volatility.

Table 5 summarises the results. For each measure, its regression coefficient (column Coeff.), standard error (Std. Err.), statistical significance (P-value) and coefficient of determination (R²) are shown. The P-value is the probability that the coefficient is different from zero by chance. Only measures that are significant at α = 0.05 should be considered for the subsequent multivariate analysis [6].
According to our data, only models 1 and 3 are significant at \( \alpha = 0.05 \), i.e. only NACTOR and NLINE should be considered for multivariate analysis. We can also see that the \( R^2 \) values of models 1 and 3 are 36.5 and 40.3, respectively, i.e. these models explain 36.5% and 40.3%, respectively, of the variation in the dependent variable. This means that NACTOR and NLINE are good predictors of volatility according to this test.

### 5.4 Multivariate regression analysis

Multivariate regression analysis was not applicable, because we discarded NACTOR as a result of the PC analysis and NWORD and NUC as a result from the univariate analysis. This leaves us only one independent variable; NLINE.

### 5.5 Sanity tests on the regression models

One of the threats to conclusion validity is the violation of assumptions of statistical tests. To be valid, the models have to satisfy some hypotheses on the residuals. We followed the tests suggested in [29]. The sanity checks below were performed on the univariate models 1 and 3. Figure 2 shows the plots generated by residuals analysis for models 1 (NACTOR) and 3 (NLINE).

1. **Linear relationship between response and predictors.**
   The “lack-of-fit-test”, did not show any evidence of lack of fit in both models for \( p \leq 0.1 \) (which is a standard threshold).

2. **Homogeneity of variance (the residuals have constant variance).** The residuals versus fitted values plots in figure 2 did not reveal any patterns for any model.

3. **Independence of residuals.** The results of the Durbin-Watson test are shown in table 6. The values obtained were higher than the upper bounds in all cases. Therefore we conclude that there is no autocorrelation; the residuals are independent.

### 5.6 Evaluating goodness of fit

In this section, we evaluate the goodness of fit of model 3 obtained in the previous paragraphs. The measures we used to evaluate the accuracy of the prediction models are the mean magnitude of relative error (MMRE) and the threshold measure \( \text{pred}(n) \). The magnitude of relative error is defined as a measure of discrepancy between the actual and the fitted values. A low MMRE generally indicates an accurate model, reliable prediction systems have MRE values between 10-30%. The MMRE can be strongly influenced by few very high relative error values. Therefore, we have included the median magnitude of relative error (MdMRE) (DFITS). From the results of these statistics, we found that model 3 did not have unusual observations, while two outliers were found for model 1. We tested if the outliers were influential because it is important that the conclusions drawn are not dependent on few outlying observations. The univariate regression model obtained by removing the two outliers from model 1 was not significant at \( \alpha = 0.05 \) (\( P=0.07 \)), i.e. the outliers were influential.

As a result of the models' checking, we decided to discard the model having NACTOR as covariate from the models' evaluation because the outliers were influential (the significance of the regression model changed).

### Table 5: Univariate regression analysis for data set A (\( \alpha = 0.05 \))

<table>
<thead>
<tr>
<th>Model</th>
<th>Measure</th>
<th>Coeff.</th>
<th>Std. Err.</th>
<th>P-value</th>
<th>( R^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NACTOR</td>
<td>0.3258</td>
<td>0.1241</td>
<td>0.022</td>
<td>0.133</td>
</tr>
<tr>
<td>2</td>
<td>NUC</td>
<td>0.0827</td>
<td>0.00506</td>
<td>0.163</td>
<td>15.6</td>
</tr>
<tr>
<td>3</td>
<td>NLINE</td>
<td>0.0071</td>
<td>0.002504</td>
<td>0.005</td>
<td>40.3</td>
</tr>
<tr>
<td>4</td>
<td>NWORD</td>
<td>0.000735</td>
<td>0.0003324</td>
<td>0.133</td>
<td>17.8</td>
</tr>
</tbody>
</table>

### Table 6: Durbin-Watson test (sample size=14)

<table>
<thead>
<tr>
<th>Model</th>
<th>Measure</th>
<th>DW value</th>
<th>Critical values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NACTOR</td>
<td>2.18504</td>
<td>[1.04495, 1.35027]</td>
</tr>
<tr>
<td>3</td>
<td>NLINE</td>
<td>2.3664</td>
<td>[1.04495, 1.35027]</td>
</tr>
</tbody>
</table>

### Figure 2: Residuals analysis for models 1 (NACTOR, at the top) and 3 (NLINE)
in the evaluation. Another measure of accuracy is \( \text{pred}(n) \) which provides an indication of overall fit for a set of data points. \( \text{pred}(n) \) is based on the MRE values for each data pair and it is defined as the percentage of the data pairs with \( \text{MRE} \leq n \). For example, \( \text{pred}(0.30) = 0.43 \) means that 43% of the fitted values fall within 30% of their corresponding actual values. The higher the \( \text{pred} \) values the more reliable is the model. Values of \( \text{pred}(0.25) \) above 0.7 mean that the system is reliable, but such performance is difficult to get [13].

Observing the \( \text{pred} \) value in table 7, our model predicts more than 90% of the cases with a relative error less than 25%. Considering a relative error of 40%, the model predicts 100% of the cases.

The ideal situation would be to compare the results obtained with other prediction models of volatility, but we have not found such models. Considering other research areas like object-oriented design and software maintenance, the goodness of fit of our model is better than those presented by Genro et al. [16] (whose best model has a MMRE=0.24) and by MacDonnel [34] (whose best model has a MMRE =0.21). Similarly to our case, these models were evaluated on the same set they were constructed. We would therefore expect to get high accuracy.

In the next section we will apply our prediction model to data set B.

6. PREDICTION MODEL VALIDATION

Our situation is ideal for evaluating prediction models, since separate data sets are available that have been derived from different projects, but within similar environments. The prediction model is built from one data set and then used to make predictions for another project. Project factors may affect the predictive power of a model and therefore it is important to validate the model under conditions that as closely as possible resemble its usage conditions [6].

The values of MMRE and of MdMRE when model 3 is applied to data set B, are shown in table 8. These values are in the range of the recommended values for reliable models. The model predicts 68% of the cases with a relative error less than 25% and it predicts 86% of the cases with a relative error less than 40%. This model performs better than COCOMO [5] (which has MMRE=0.6 and \( \text{pred}(0.25)=0.27 \)) and Jörgensen’s best model [25] (MMRE=1.0 and \( \text{pred}(0.25)=0.26 \)).

It is interesting to note that our model depends on the length of the requirements and not on the functionality or complexity. This result is in accordance with the principal component analysis, where the PC1 was the most influential.

7. THREATS TO VALIDITY

A discussion of possible threats to validity will help us to qualify the results and highlight some of the issues associated with our study.

Conclusion validity. One issue that could affect the conclusion validity is the relative small size of the sample data. Concerning data quality, the data for NLINE and NWORD have been collected using a computerised tool and are therefore reliable. The data collection for NCHANGE, NUC, NACTOR, and NREVISION involved human judgement. However, we have defined measurement rules to keep the judgment as objective as possible (see appendix A). Furthermore, the measurement rules have been tested; three subjects have independently applied the rules on two versions of a requirement document obtaining the same results. In retrospective studies there is a risk of incomplete data. In our case all files were obtained from the companies’ version handling system. The project manager acknowledged that all requirements documents were handled by this system and we could not find any dangling file reference.

Construct validity. The construct validity is the degree to which the variables are accurately measured by the measurement instruments used in the study. The construct validity of the measures used for the independent and dependent variables, is ensured by the theoretical validation performed in 4.4.

External validity. Our threats to external validity are minimal because the study was performed in an industrial environment, therefore the materials used, and the projects themselves were real. Other factors that may affect the external validity of this study could be the project length, the experience of customers and developers in specifying requirements, the techniques used to collect and document requirements, and the company’s maturity (in requirements and software processes). The projects were developed in the same company and used (approximately) the same processes, but everything else was different: size, personnel, and type of project. That means, we can assume that the models are valid for a relatively large class of projects within the company.

Internal validity. The study described here is a correlational study. We have shown that all measures are significantly correlated to Volatility as defined in section 4.3. However, this relationship does not imply a causal relationship. Only a controlled experiment, where the measures would be varied in a controlled manner (and the other factors would be held constant), could demonstrate causality, as we have discussed in section 4.5. One threat to internal validity is the potential inaccuracy of collected data on changes to requirements. Change requests were not available. Actual changes to requirements documents were determined by comparing all revisions of the requirements documents analysed. Rules of measurements on how to count changes were defined (see appendix A). However, we had access to and analysed all revisions of the requirements documents analysed. Version
and configuration management is handled seriously at the company (see conclusion validity).

8. DISCUSSIONS AND CONCLUSIONS

In this paper, we have described a correlational study on requirements volatility performed at BAE Systems Hägglunds AB, Sweden. We collected and analysed data of two historical projects in the company for the measures NUC, NACTOR, NLINE, NWORD, NREVISION, and NCHANGE. Applying univariate and multivariate regression analysis, we built prediction models using data collected on a medium size software project. Only the model having NLINE as covariate was found significant. We then evaluated the model accuracy by applying it on a set of data collected on a second, slightly larger, project developed at the same company.

Our hypothesis was: the size measures NACTOR, NUC, NWORD, and NLINE, are good predictors of Volatility (as defined in section 4.3). The data analysis showed that the measure NLINE has a significant positive relationship to volatility. The accuracy of the measure NLINE as predictor is shown by the low value of MMRE and high value of pred(n) (see table 7). The other measures were not significant predictors. As a result of the PC analysis and of the univariate analysis we could not combine the independent variables to construct a multivariate model. When applying model 3 on data set B, NLINE has good performance, the values of MMRE obtained are in the range of the recommended values for reliable models.

Our prediction model receives as input the number of lines of a file describing software requirements. The model produces a number as output: the sum of the change densities in time for this requirements document. By regularly comparing the number obtained (the predicted volatility) to the current volatility (as suggested by Costello and Liu [12]), project managers can identify critical requirements and allocate resources for analysing reasons for their volatility. In this way they can minimise the risks of schedule and cost overruns.

In the present work, we deal with two factors of volatility: the number of changes to requirements and time. As discussed in sections 4.2 and 4.3, volatility is a quite complex concept depending on many more factors than size. For a deeper analysis of volatility it is suggested to investigate qualitative aspects such as why the changes occur, how critical the changes are, the type and phase where the changes occur. Regularly investigating many qualitative aspects on many requirements documents is, however, expensive and subjective and therefore not feasible. However, studying the impact of a change might help to “classify” changes and to identify the most critical changes. In a qualitative analysis, we would care only about important changes since all others will not affect the project much. With our prediction model we can quantify the “instability” of requirements in order to identify the critical ones. When the critical requirements are identified, we can perform a deeper analysis of the changes in order to figure out the problems with the requirement.

Although our approach currently uses use case-based requirements, the defined measures are quite general and can therefore be used for all use case documents written in textual form. Furthermore, the measure NLINE can be applied to any kind of requirement written in textual form. Our approach of using counts is simple, effective, easily interpreted, and completely automated.

We believe it is possible to develop prediction systems based on simple measures such as ours. The reason for our confidence is that the correlational study presented here is based on two different industrial projects. Although the exact nature of the prediction systems will vary from company to company, the underlying principle is the same. That is, developers can collect data for simple measures derived from requirements documents, and build effective prediction systems using techniques like linear regression analysis. Models must however be calibrated to suit different environments [34].

We plan to apply the models in larger projects in the same company and in other companies. The study described in this paper needs to be replicated in a variety of environments and systems in order to build a body of knowledge in the area.

9. REFERENCES


APPENDIX
A. MEASUREMENT RULES
In this section, we describe the rules we adopted for measuring the requirements documents of the two analysed projects. For this study we considered files as units of requirements documents. We had full access to the repository of the projects and retrieved all existing revisions of all files analysed from the start of the projects until the point in time of the analysis.

The measure $N\text{CHANGE}$ was obtained by counting the number of changes to a unit of requirement document. We compared two successive versions of a file using the “track changes: compare documents” tool in MS Word. Each word added or deleted was considered as one change, while each word substituted was considered as two changes (one deletion plus one addition). Furthermore, adding a picture, resizing, adding a frame, adding or deleting a detail in a picture were all considered as one change. Intuitively, adding a picture should be considered a bigger change compared to resizing it. However, if changes are made in the pictures these are usually mirrored with changes in the text.

Exceptions to these rules were the following. We did not count addition or deletion of empty space, empty lines added or deleted, page breaks or tabs without text. Automatic changes like the date in the headers and the filename in the footers were not considered changes. If a section was inserted or deleted and the successive section’s numbers were changed, they were not considered as changes. The same change repeated many times in different paragraphs was considered as many changes. There were changes like a substitution of a newline where the word after the newline is highlighted in red. We did not considered these words as changed.