Construction and Validation of Prediction Models for Number of Changes to Requirements

Annabella Loconsole & Jürgen Börstler

Department of Computing Science, Umeå University, SE-90187 Umeå, Sweden
{bella,jubo}@cs.umu.se, Phone:+46 90 7866372, Fax:+46 90 7866126

Abstract

In this paper we present a correlational study in which we assess the ability of five size measures to predict the number of changes to requirements 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 of number of changes. 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 projects for five measures of size of requirements (number of actors, use cases, words, lines, and revisions), we have built and evaluated prediction models for number of changes to requirements. 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. We performed a cross systems validation. For our best model we calculated a pred(0.25)=0.5, which is better than the accuracy of prediction models like COCOMO. 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 developer perception are unreliable. Predictions models, like the one presented here can help to mitigate that risk.

Key words: Requirement, Prediction Model, Empirical Validation, Correlational Study

1 Introduction

Requirements engineering is an important phase of software development, where the needs of the stakeholders are collected, developed, and documented. Requirements development is a learning rather than a gathering process. As a consequence, requirements change frequently, even during later stages of the
development process. Software requirements that change often are usually said to be volatile. Studies show that requirements volatility has a high impact on project performance (Pfahl and Lebsanft, 2000; Stark et al., 1999; Zowghi and Nurmuliani, 2002). However, we cannot expect the requirements to be stable, even when requirements engineering tasks (such as elicitation, analysis, specification, and validation) are well performed. It is therefore important to carefully monitor and control the requirements throughout the software life cycle. Monitoring requirements volatility usually involves measuring trends or percentages of changes to requirements (see section 4.2). Anticipating a certain level of volatility project managers can take appropriate actions in order to decrease project risks.

In this paper, we describe a correlational study with the goal of empirically validating five measures of requirements size as predictors for the number of requirements changes. We built seven prediction models using data collected for a medium-size software project developed at BAE Systems Hägglunds AB, Sweden. We then evaluated the accuracy of five models by applying them on a set of data collected for a second project at the same company.

The results show that the best predictors of number of changes are the length measures: number of lines and words. Other predictors of complexity and functionality were found less accurate. The models created can be used on a timely basis to predict volatility trends.

The remaining part of the paper proceeds as follows: section 2 describes the research related to empirical validation of measures in general and work related to requirements volatility. In section 3 we briefly summarise our previous case study, which investigated the relationship between four measures of size of use case models and requirements volatility. Section 4 describes the goals, hypotheses, and data collected in the present empirical study. The data analysis and the resulting prediction models are described in section 5. Finally, discussions and conclusions are presented in section 6.

2 Related work

There is little empirical research in the area of requirements volatility. The majority of the published studies, evaluate the impact of requirements volatility on software projects (Stark et al., 1999; Zowghi and Nurmuliani, 2002), on software products (Henry and Henry, 1993), and defect density (Javed et al., 2004; Malaiya and Denton, 1999).

Many measures related to requirements volatility have been proposed in the
Measures to assess requirements stability\footnote{Sometimes, the word stability is used instead of volatility. For instance, a definition of “degree of stability” of requirements is presented in (IEEE830-1998, 1998). Both terms are used conjunctly in (Huffman et al., 1998; Raymus, 1999). In our opinion, the words are antonyms.} are presented by Ambriola and Gervasi (2000). They showed that requirements stability had a high predictive value of project risks in a requirements analysis process. Other measures (Costello and Liu, 1995; Henderson-Sellers et al., 2002; Huffman et al., 1998; Hyatt and Rosenberg, 1996; Malaiya and Denton, 1999; Nurmuliani et al., 2004; Raymus, 1999; Stark et al., 1999) concern requirements volatility. In most of the cases, the definition of volatility expresses the changing nature of the requirements (see section 4.2 for further details). A few correlational studies are present in the requirements volatility literature (Ambriola and Gervasi, 2000; Henry and Henry, 1993; Javed et al., 2004; Loconsole and Börstler, 2005; Stark et al., 1999) but, except Loconsole and Börstler (2005), requirements volatility was chosen as independent variable i.e. as predictor of other attributes. Henry and Henry (1993) for example, propose measures to predict the impact of volatility on the software product. Similarly, Stark et al. (1999) present measures to predict the effects of changing requirements on costs and schedule. Javed et al. (2004) present measures of correlation between volatility and software defects.

The majority of the measures above are designed for well specified and well written requirements, using standardised documentation templates. However, even in the case of well-documented requirements, measures have to be tailored towards the particular organisation, because each company has its own way of documenting requirements. Among those measures, only Ambriola and Gervasi (2000) and Loconsole and Börstler (2005) provide an empirical validation. The reason for validating measures is to empirically demonstrate their practical utility, i.e. to show that there is a consistent relationship between the measure and an external attribute (Fenton and Pfleeger, 1996; Kitchenham et al., 1995; Schneidewind, 1992; Zuse, 1997). It is important to ensure that the data collected for a certain measure is related to the actual property investigated (the attribute to be measured). Otherwise time and money is spent for collecting useless data. To our knowledge, only one empirical validation of requirements volatility measures has been performed in an industrial setting (Loconsole and Börstler, 2005). In that study we showed a high correlation between four measures of size and number of changes to use case models (see section 3 for further details).

The measures we are interested in are measures that can help to predict the number of changes to requirements and this, in turn, can be used to determine volatility. There are no studies on prediction models of requirements changes. Bush and Finkelstein (2002, 2003) describe a process that could sup-
port a predictive view of requirements stability. Starting from an initial set of requirements, this process helps to create worlds of possible evolutions of requirements. They also report on positive results from an industrial case study validating the approach. However, using their process is complex and time consuming. Our approach is much simpler and better suited for small and medium size companies. It is based on use case requirements. Nevertheless, except use case diagrams, the requirements are mainly text based, therefore, it is possible to generalise our results. Further studies are needed to prove this.

3 Background

In an earlier industrial case study (Loconsole and Börstler, 2005), we investigated measures of volatility for a medium size software project. Our goals were: 1) to empirically validate a set of measures associated with the volatility of requirements documents; and 2) to investigate the correlation between perceived and measured volatility. We collected size and change data in retrospect for all versions of requirements documents of the software project. In addition, we determined the perceived volatility by interviewing stakeholders of the project.

The spearman correlation coefficient was calculated between each measure of size of requirements documents and the size of changes 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 our measures of size of requirements documents are good indicators of the number of changes for use case based requirements documents. For the second goal, 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. These results suggest that managers at this company should measure their projects because of the risk to take wrong decisions based solely on their own and the developers perceptions.
### Table 1

Key data of the two projects analysed.

<table>
<thead>
<tr>
<th></th>
<th>Project A</th>
<th>Project B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of developers</td>
<td>10 (incl. 2 proj. managers)</td>
<td>15</td>
</tr>
<tr>
<td>Project duration</td>
<td>30 months</td>
<td>48 months</td>
</tr>
<tr>
<td>Use case documents (UCD)</td>
<td>14 files</td>
<td>22 files</td>
</tr>
<tr>
<td>Average UCD size in kB</td>
<td>Mean 65, Median 60</td>
<td>Mean 101, Median 76.5</td>
</tr>
<tr>
<td>Total number of req. documents</td>
<td>16</td>
<td>26</td>
</tr>
<tr>
<td>Vision/overview documents</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Supplementary req. documents</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Total number of use cases</td>
<td>39</td>
<td>22</td>
</tr>
</tbody>
</table>

### 4 Description of the correlational study

The goal of the present study is to analyse the ability of five specific measures to predict the number of changes to requirements, using two data sets. Prediction models were constructed applying univariate and multivariate linear regression to the data set from our previous study (see section 3 and Loconsole and Börstler (2005)). The models were then validated using a second new dataset. The data collection was semi-automatic, carried out by the authors by studying historical project’s documentation. We analysed the files starting from the first available revision, 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 (Briand and Wüst, 2002).

#### 4.1 Context of the study

We analysed and collected data from the requirements specifications 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 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 Loconsole and Börstler (2005) for the UCM template utilised in project A). Use case modelling was the only technique used in this project for describing func-
tional requirements. The vision document contained only sketchy, high level requirements and was not analysed. 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 documents. According to the project 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”. In these projects, we consider use cases as requirements.

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.

4.2 Dependent variables

Our goal was to investigate the relationship between requirements size measures and requirements volatility. We therefore had to choose a suitable and practical measure of volatility as the dependent variable of our study. Theoretical definitions of requirements volatility are presented in (Baumert and McWhinney, 1992; Nurmuliani et al., 2004; Raynus, 1999; Rosenberg and Hyatt, 1996), while operational definitions can be found in (Baumert and McWhinney, 1992; Chrissis et al., 2003; Hyatt and Rosenberg, 1996; Locomsole and Börstler, 2005; Nurmuliani et al., 2004; Raynus, 1999; Stark et al., 1999). Baumert and McWhinney (1992), suggest to measure source and state of change, while Nurmuliani et al. (2004) take into consideration the source of change in their theoretical definition of volatility. Except these two cases, all definitions have several things in common:

(1) They express the changing nature of requirements during the software development.

We are aware of the fact that some researchers do not consider use cases as requirements (Ham, 1998; Lau, 2004; Schneider and Winters, 1998; Young, 2004).
(2) They focus on the amount of changes (additions, deletions, and modifications) to requirements.
(3) They do not consider the cause of change and the semantics of a change, i.e. in what way a change impacts development.

That means that volatility is treated as a quantitative measure. Likewise, we define requirements volatility as the amount of changes to a requirements document over time, and will measure it as the number of changes (NCHANGE) to a requirements document. There is one difference between our operational definition of volatility and the ones above. We look at volatility document by document instead of treating all requirements as one set. This enables us to identify different degrees of volatility within the whole set of requirements.

NCHANGE is a direct measure which is necessary in order to calculate other important indirect measures like change density and frequency of change. Requirements change density can easily be computed by dividing NCHANGE by requirements size. Frequency of change is calculated by applying NCHANGE within specific time intervals. In this way we can also identify volatility trends.

Please note that we do not do any cause-effect or impact analyses 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. The downstream artefacts that could possibly be affected by a change (design and code for example) are not available yet.

Our dependent variable NCHANGE has been determined by comparing versions of requirements documents by means of a tool and counting the changes from one version of a document to the next. A detailed description of the counting rules can be found in appendix A.

4.3 Independent variables

The choice of the independent variables depends on the entity and the size of the systems measured. Because the projects under analysis are different from each other, it is necessary to select general measures that can be applied in both project contexts. This is also necessary to increase general applicability of results.

The entities analysed in the two projects were requirements documents. In project A we analysed fourteen files each containing a small use case model. In project B we analysed twenty-two files, each containing one use case. Intuitively, the larger the document the more changes there are. Therefore, we believe that the size of requirements is the most influential factor affecting
volatility. The size measures “number of actors interacting with the use cases described in the file” (NACTOR), “number of lines per file” (NLINE), “number of words per file” (NWORD), “number of use cases per file” (NUC), and “number of revisions per file” (NREVISION) are the independent variables chosen for this study. As suggested by Fenton and Pfleeger (1996), size can be seen as composed of length, functionality, and complexity. In our case, NLINE and NWORD are measures of length, NACTOR and NREVISION are measures of complexity, and NUC is a measure of functionality. These measures are quite intuitive. The NLINE and NWORD are simply a count of lines and words of the files analysed and were calculated by the authors using a computerised tool. NACTOR is a count of the number of actors interacting with the use cases described in each file analysed. NREVISION is a count of the revisions for each file. A revision is a version of a file with a unique identifier. In our previous study (see section 3) we did not analyse the correlation of NREVISION with number of changes.

The independent variables are defined as measures of size. The size of a requirement document can be computed at varying levels of granularity, because the requirements documents are organised hierarchically. We did not collect measures at higher or lower abstraction level, because we considered those requirements as either too vague or too close to the design level.

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 Armour (2004), what we actually want to measure is how much knowledge there is in our system or file. Unfortunately, there is not yet an empirical way to measure knowledge. A possible choice for the independent variable could be use case points (UCPs) (Schneider and Winters, 1998). Effort estimation models based on UCP have been investigated by Anda (2002). However, UCPs are not generally applicable. The definition of UCPs is based on a classification of use cases and a number of environmental factors (similar to the cost drivers in COCOMO (Boehm, 1981)). This information was not available for our projects. The classification of use cases is a subjective activity, therefore it is not possible to collect UCPs automatically.

Other possible independent variables could be the total number of requirements, the number of requirements added and deleted, or the number of initial and final requirements. These are system measures, i.e. we need to have several systems to be able to count these measures. Furthermore, these measures would make it necessary to exactly define what a single atomic requirement is, otherwise it cannot be counted reliably. Other measures like “number of associations between use cases”, “number of steps in scenarios”, were discarded because they were not readily available. The measures chosen for this study are not necessarily the most appropriate for all projects but were a suitable
4.4 Hypothesis

The hypothesis of the study is the following: the size measures NACTOR, NUC, NWORD, NLINE, and NREVISION are good predictors of number of requirements changes. Our hypothesis is built on the idea that larger requirements are affected by changes more than smaller ones, because they contain more information. The relationship between number of changes and our five measures of size 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 number of changes to requirements. The study does not make any claims with respect to causality which can be proven only by performing controlled experiments. In industrial environments it is usually hard to perform controlled experiment. For real projects, 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, and the company’s maturity (in requirements and software processes in general). In academic environments, on the other hand, the small size of the projects does not usually allow to check for changes to requirements (Loconsole and Börstler, 2004).

5 Analysis results

As described in section 3, in our previous study we found a strong correlation between four of the five size measures introduced in section 4.3 with total number of changes (Loconsole and Börstler, 2005). Furthermore, the scatter plots of our measures versus number of changes show approximately a linear correlation (see figure 1 and 2), especially for the length measures. Based on those results, we have analysed the ability of our measures to predict the number of changes to requirements. For this analysis we applied univariate and multivariate linear regression which is suggested to predict interval and ratio scale dependent variables (Briand and Wüst, 2002).

The data analysis is obtained by following the procedure suggested by Briand and Wüst (2002); Briand et al. (2000), which is also described in statistical books such as Draper and Smith (1966). We start with the descriptive statistics performed on data sets A and B (section 5.1). The principal component analysis (5.2), univariate 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 5.7 we evaluate the
prediction models by applying them on data set B.

5.1 Descriptive statistics

Table 2 shows the descriptive statistics for data sets A and B. The columns SE Mean, StDev, and IQR state respectively mean standard error, standard deviation, and inter quartile range.

In both data sets, the mean NCHANGE is larger than the median\(^3\). The standard deviation in data set B is higher than the mean, this is due to some outliers. Furthermore, the mean is more than two times higher than the mean in data set A. This implies that the prediction models, built based on data set A, will probably predict low numbers of changes for data set B.

![Scatterplot of number of changes (project A)](image)

Fig. 1. Scatter plot project A

NACTOR, NUC, and NREVISION have relatively low means, standard deviations, and variances in both data sets. Low variance measures do not differentiate entities very well, therefore they are not likely to be useful predictors. The low variance of NACTOR and NUC is due to the fact that the requirements documents under analysis contain very few use cases and actors. However, NACTOR and NUC can be expected to be low, since UCMs should be kept

\(^3\) Extremely 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 (Zar, 1999).
small according to common guidelines (Jacobson, 2004; Lilly, 1999). The totals of NACTOR in the two data sets are comparable, while the range and the median are very different. This means that the 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 in project B each file describes only one use case (except one file which described two use cases). According to Briand et al. (2000), only measures with more than five non zero data points should be considered for further analysis. This means that NUC might be a useless measure when we have one use case per file. In this case a better measure might be number of use cases per UCMs.

NWORD has the largest mean and standard deviation in data set A. The mean values of NWORD and NLINE in data set B are larger than the median, due probably to some outliers.

Besides some exceptions (NUC and NREVISION), the ranges of the measures in data set B are larger than in data set A. The totals are in the order of two times of data set A. 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. The subsequent analysis will be done on data set A. After that, the constructed prediction models will be applied on data set B for evaluation.

Fig. 2. Scatter plot project B
### Table 2

Descriptive statistics for data sets A and B.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Measures</th>
<th>Range</th>
<th>Total</th>
<th>Mean</th>
<th>SE Mean</th>
<th>StDev</th>
<th>Variance</th>
<th>Median</th>
<th>IQR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
<td>NCHANGE</td>
<td>900</td>
<td>5360</td>
<td>382.9</td>
<td>73.3</td>
<td>274.1</td>
<td>75138.9</td>
<td>303.5</td>
<td>345</td>
</tr>
<tr>
<td></td>
<td>NACTOR</td>
<td>2</td>
<td>5</td>
<td>1.5</td>
<td>0.174</td>
<td>0.65</td>
<td>0.423</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>NUC</td>
<td>5</td>
<td>39</td>
<td>2.786</td>
<td>0.447</td>
<td>1.672</td>
<td>2.797</td>
<td>3</td>
<td>2.25</td>
</tr>
<tr>
<td></td>
<td>NLINE</td>
<td>105</td>
<td>1436</td>
<td>102.57</td>
<td>8.35</td>
<td>31.24</td>
<td>975.96</td>
<td>100.5</td>
<td>52.5</td>
</tr>
<tr>
<td></td>
<td>NWORD</td>
<td>852</td>
<td>8771</td>
<td>626.5</td>
<td>73.9</td>
<td>276.4</td>
<td>76375.5</td>
<td>663.5</td>
<td>492.8</td>
</tr>
<tr>
<td></td>
<td>NREVISION</td>
<td>5</td>
<td>89</td>
<td>6.357</td>
<td>0.44</td>
<td>1.646</td>
<td>2.709</td>
<td>6.5</td>
<td>3.25</td>
</tr>
<tr>
<td><strong>B</strong></td>
<td>NCHANGE</td>
<td>3600</td>
<td>17689</td>
<td>804</td>
<td>191</td>
<td>895</td>
<td>801152</td>
<td>395</td>
<td>975</td>
</tr>
<tr>
<td></td>
<td>NACTOR</td>
<td>5</td>
<td>6</td>
<td>3.182</td>
<td>0.276</td>
<td>1.296</td>
<td>1.68</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>NUC</td>
<td>1</td>
<td>23</td>
<td>1.0455</td>
<td>0.0455</td>
<td>0.2132</td>
<td>0.0455</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NLINE</td>
<td>378</td>
<td>2848</td>
<td>129.5</td>
<td>18.7</td>
<td>87.9</td>
<td>7720.1</td>
<td>109</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>NWORD</td>
<td>2379</td>
<td>16831</td>
<td>765</td>
<td>123</td>
<td>578</td>
<td>333794</td>
<td>606</td>
<td>822</td>
</tr>
<tr>
<td></td>
<td>NREVISION</td>
<td>7</td>
<td>64</td>
<td>2.909</td>
<td>0.354</td>
<td>1.659</td>
<td>2.753</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

### Table 3

Rotated components for data set A.

<table>
<thead>
<tr>
<th>Measures</th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>PC4</th>
<th>PC5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eigenvalue</td>
<td>3.3979</td>
<td>0.7848</td>
<td>0.6296</td>
<td>0.1732</td>
<td>0.0145</td>
</tr>
<tr>
<td>Percent</td>
<td>68</td>
<td>15.7</td>
<td>12.6</td>
<td>0.35</td>
<td>0.03</td>
</tr>
<tr>
<td>Cumulative</td>
<td>68</td>
<td>83.7</td>
<td>96.2</td>
<td>99.7</td>
<td>100</td>
</tr>
<tr>
<td>NACTOR</td>
<td>0.308</td>
<td>-0.183</td>
<td><strong>-0.929</strong></td>
<td>0.091</td>
<td>-0.003</td>
</tr>
<tr>
<td>NLINE</td>
<td><strong>0.807</strong></td>
<td>-0.290</td>
<td>-0.385</td>
<td>0.321</td>
<td>-0.115</td>
</tr>
<tr>
<td>NUC</td>
<td>0.496</td>
<td>-0.328</td>
<td>-0.112</td>
<td><strong>0.796</strong></td>
<td>-0.005</td>
</tr>
<tr>
<td>NREVISION</td>
<td>0.128</td>
<td><strong>-0.954</strong></td>
<td>-0.175</td>
<td>0.207</td>
<td>-0.008</td>
</tr>
<tr>
<td>NWORD</td>
<td><strong>0.92</strong></td>
<td>-0.051</td>
<td>-0.257</td>
<td>0.285</td>
<td>0.055</td>
</tr>
</tbody>
</table>

### 5.2 Principal component analysis

Principal component (PC) analysis is performed in order to form a smaller number of uncorrelated variables, by selecting the independent variables with high loadings. 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 3 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.

As we can see in table 3, NLINE and NWORD get high factor loadings in PC1. They express the same orthogonal dimension, i.e. the length of requirements documents. NREVISION, NACTOR, and NUC get high loadings in PC2, PC3, and PC4, respectively. PC2 and PC3 express complexity, while PC4 expresses functionality dimensions. Among the five PCs shown in the table, three of them show sufficient variance, as shown by the scree plot in figure 3.
The three PCs with high loadings capture 96.2% of the variance in the data set. PC4 and PC5 account for too small percentage of variance and should therefore be eliminated. Thus the measure NUC 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 4. Regression analysis is conducted here to investigate the importance of each of the five size measures (the independent variables) in determining the number of changes (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 five measures (also called ordinary least squares regression), which is most suitable to predict a dependent variable at the interval and ratio scale (Briand and Wüst, 2002). Each of the measures was found to have a statistically significant positive relationship with NCHANGE. Table 4 summarises the results. For each measure, the regression coefficient, its standard error, the coefficient of determination ($R^2$), and the statistical significance (P-value) are shown. The P-value is the probability that the coefficient is different from zero by chance. Only measures that are significant at $\alpha = 0.05$ should be considered for the subsequent multivariate analysis (Briand and Wüst, 2002).

Model 2 has NUC as covariate, which had low variance in data set A (see table 2). This model has also the highest P-value and lowest $R^2$ (even though it is still significant). Similarly, model 5 has low $R^2$. It also has one influential outlier (the outlier is influential if the significance of the model changes when
performing the test without the outlier). Therefore, this model will be evaluated without the outlier (as suggested by Briand and Wüst (2002)). A deeper analysis of outliers and model checking is done in section 5.5.

Observing table 4, the $R^2$ value of model 3 is 80.2. This means that model 3 explains 80.2% of the variation in number of changes. Similarly we can interpret the results for the other models. The goal of this test was to determine if each measure is a useful predictor of number of changes. Although the measures are all significant, NACTOR, NLINE, and NWORD seem to be better predictors than NUC and NREVISION.

5.4 Multivariate regression analysis

In this section we present the construction of prediction models built on data set A, with the goal of accurately predicting the number of changes to requirements. Because this study is exploratory, we do not know which independent variables should be included in the prediction models. Usually, a stepwise selection process is used (Briand and Wüst, 2002; Levine et al., 2001; Zar, 1999). A common method to reduce the number of independent variables is to use the results from the principal components analysis as filter, selecting only the variables with high loadings in the significant PCs. In our case only three PCs had sufficient variance, and this lead us to discard the measure NUC. We applied the multivariate linear regression starting with four variables. The statistics tool used for the analysis (Minitab) gives the possibility to check for collinearity. As expected, NLINE and NWORD were found collinear, therefore we continued the statistical analysis excluding NLINE.

Only two models with two covariates were found significant (see table 5). Both models have NWORD as one of the covariates. Model 6 (NWORD and NREVISION) had one influential outlier. Therefore, in the following sections, it will be evaluated without the outlier (as suggested by Briand and Wüst (2002)). A deeper analysis of outliers and model checking is done in section 5.5. We discarded the models with a P-value higher than 0.05 in the t-test.
Table 5
Multivariate regression models for data set A.

<table>
<thead>
<tr>
<th>Models</th>
<th>Measures</th>
<th>Coeff.</th>
<th>Std. Err.</th>
<th>P-value</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>NWORD</td>
<td>0.6148</td>
<td>0.1239</td>
<td>0.001</td>
<td>82.6</td>
</tr>
<tr>
<td></td>
<td>NREVISION</td>
<td>64.39</td>
<td>21.2</td>
<td>0.013</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Intercept</td>
<td>-444.7</td>
<td>137.9</td>
<td>0.009</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>NACTOR</td>
<td>187.19</td>
<td>75.4</td>
<td>0.030</td>
<td>75.7</td>
</tr>
<tr>
<td></td>
<td>NWORD</td>
<td>0.5362</td>
<td>0.1775</td>
<td>0.012</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Intercept</td>
<td>-233.9</td>
<td>112.6</td>
<td>0.062</td>
<td></td>
</tr>
</tbody>
</table>

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 hypothesis on the residuals. We followed the tests suggested in (Levine et al., 2001). The sanity checks were performed on five univariate models and two multivariate models. See figure 4 for an example of the plots generated for the analysis of the residuals.

Fig. 4. Residuals analysis for model 2, NUC

The results of the model checking is listed below.

1. **Linear relationship between response and predictors.** The "lack-of-fit-test", did not show any evidence of lack of fit for \( p \geq 0.1 \) in any of the seven models.

2. **Homogeneity of variance (the residuals have constant variance).** The residuals versus fits plot did not reveal patterns in any case.
Table 6: Durbin-Watson test

<table>
<thead>
<tr>
<th>Models</th>
<th>Measures</th>
<th>DW values</th>
<th>Critical values</th>
<th>Sample size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NACTOR</td>
<td>1.78</td>
<td>[1.04495, 1.35027]</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>NUC</td>
<td>2.46</td>
<td>[1.04495, 1.35027]</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>NLINE</td>
<td>2.12</td>
<td>[1.04495, 1.35027]</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>NWORD</td>
<td>2.14</td>
<td>[1.04495, 1.35027]</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>NREVISION</td>
<td>2.6</td>
<td>[1.00973, 1.3404]</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>NWORD, NREVISION</td>
<td>1.65</td>
<td>[0.86124, 1.56212]</td>
<td>13</td>
</tr>
<tr>
<td>7</td>
<td>NWORD, NACTOR</td>
<td>2.31</td>
<td>[0.90544, 1.55066]</td>
<td>14</td>
</tr>
</tbody>
</table>

(3) Independence of residuals. The results of the Durbin-Watson test are shown in table 6. The values we obtained were higher than the upper bound in all cases, therefore we conclude that there is no autocorrelation, the residuals are independent.

(4) Normality of the residuals. The normal probability plot show an approximately linear pattern consistent with a normal distribution. In all cases, except for NREVISION, there is a normal distribution of residuals.

(5) No unusual observations or outliers. Minitab provides statistics tools which facilitates the detection of influential points such as leverages values, Cook’s distance, and adjusted difference (DFITS). From the results of these statistics, we found that model 1 has two unusual observations. Models 2, 3, 5, and 6 have one unusual observation while 4 and 7 have no outliers. We tested if the outliers found were influential because it is important that the conclusions drawn are not dependent on few outlying observations. Analysing the models without the outliers, we obtained that all the models were still significant at $\alpha = 0.05$.

After the analysis of the residuals, we decided to discard models 2 and 5 from the models’ evaluation because the residuals were not normally distributed and the models have the lowest performance. Furthermore, model 2 cannot be applied to data set B, due to the very low variance of NUC in data set B.

5.6 Evaluating goodness of fit

In this section, we evaluate the goodness of fit of three univariate models and two multivariate models obtained in the previous paragraphs. For prediction models constructed using the ordinary least square, the goodness of fit is measured with $R^2$. According to Briand and Wüst (2002), “while this measure allow, to some degree, for comparison of accuracy between studies, such measure is an abstract mathematical artifact that does not very well illustrate the potential benefits of using the prediction models for decision making”. The measures suggested to evaluate the accuracy of the prediction models are the following: mean magnitude of relative error (MMRE), the threshold measure
Table 7

Evaluation of the goodness of fit

<table>
<thead>
<tr>
<th>Models</th>
<th>Measures</th>
<th>MMRE</th>
<th>MdMRE</th>
<th>Pred(0.25)</th>
<th>Pred(0.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NACTOR</td>
<td>0.711</td>
<td>0.312</td>
<td>0.43</td>
<td>0.71</td>
</tr>
<tr>
<td>3</td>
<td>NLINE</td>
<td>0.2774</td>
<td>0.2729</td>
<td>0.5</td>
<td>0.93</td>
</tr>
<tr>
<td>4</td>
<td>NWORD</td>
<td>0.3903</td>
<td>0.2858</td>
<td>0.43</td>
<td>0.78</td>
</tr>
<tr>
<td>6</td>
<td>NWORD NREVISION</td>
<td>0.1757</td>
<td>0.1150</td>
<td>0.64</td>
<td>0.86</td>
</tr>
<tr>
<td>7</td>
<td>NWORD NACTOR</td>
<td>0.3852</td>
<td>0.3598</td>
<td>0.28</td>
<td>0.86</td>
</tr>
</tbody>
</table>

pred(n), correctness, and completeness (Briand and Wüst, 2002). 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 systems have MRE 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) in the evaluation.

Another measure of accuracy is pred(l) which provides an indication of overall fit for a set of data points. Pred(l) is based on the MRE values for each data pair and it is defined as the percentage of the data pairs with MRE <= l. For example, pred(0.30) = 0.43 means that 43% of the fitted values fall within 30% of their corresponding actual values. The higher the pred values the more reliable is the model. Values of pred(0.25) above 0.7 mean that the system is reliable, but such performance is difficult to get (DeLucia et al., 2005).

As we can observe in table 7, the values of MMRE are between 17-71%. Models 3 and 6 have MMRE between 10 and 30 percent, this indicate that the models are reliable. In particular, model 6 has the lowest MMRE. Models 1 and 4 have higher MMRE but the MdMRE is close to 30%. Observing the pred values, no one of our models reach a pred(0.25) = 0.7 as suggested for reliable models. However, models 3 and 6 predict more than 50% of the cases with a relative error less than 25%. Considering a relative error of 50%, models 3 and 6 can predict more than 85% of the cases.

The ideal situation would be to compare the results obtained with other prediction models of number of requirements changes, but we have not found such models. Considering other research areas like object oriented design and software maintenance, the accuracy of our models is comparable with those presented in Genero et al. (2003) (whose best model has a MMRE=0.24) and MacDonell (1997) (whose best model has a MMRE=0.21). It must be noted that the prediction models in table 7 were applied to the same data set they were derived from. We would therefore expect to get high accuracy. In the next section we will apply our prediction models to data set B.
Table 8
Validation of the prediction models

<table>
<thead>
<tr>
<th>Models</th>
<th>Measures</th>
<th>MMRE</th>
<th>MdMRE</th>
<th>Pred(0.25)</th>
<th>Pred(0.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NACTOR</td>
<td>1.518</td>
<td>0.599</td>
<td>0.23</td>
<td>0.45</td>
</tr>
<tr>
<td>3</td>
<td>NLINE</td>
<td>0.576</td>
<td>0.272</td>
<td>0.5</td>
<td>0.63</td>
</tr>
<tr>
<td>4</td>
<td>NWORD</td>
<td>0.464</td>
<td>0.3935</td>
<td>0.14</td>
<td>0.77</td>
</tr>
<tr>
<td>6</td>
<td>NWORD NREVISION</td>
<td>0.869</td>
<td>0.808</td>
<td>0.045</td>
<td>0.23</td>
</tr>
<tr>
<td>7</td>
<td>NWORD NACTOR</td>
<td>0.88</td>
<td>0.502</td>
<td>0.32</td>
<td>0.5</td>
</tr>
</tbody>
</table>

5.7 Application of the prediction models to data set B

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 models are 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 (Briand and Wüst, 2002).

As we can observe from table 8, the values of MMRE are in the range 46-151%, when the models are applied to data set B. These values are not similar to the recommended values for reliable models. This is due to the differences between the data sets (see section 5.1). However, the best models (3 and 4) have MMRE between 40 and 60%. Furthermore, model 3 predicts more than 50% of the cases with a relative error less than 25%. This model performs better than COCOMO (Boehm, 1981) (which has MMRE=0.6 and Pred(0.25)=0.27) and Jørgensen’s best model (Jørgensen, 1995) (MMRE=1.0 and Pred(0.25)=0.26). Model 6 which had the lowest MMRE in data set A does not perform very well on data set B, it seems to be fitted for dataset A. Model 3 seems to be a stable model, it has a good performance on both data sets.

It is interesting to note that the univariate models have higher accuracy compared to the multivariate models. Furthermore, the best models are those which depend 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.

5.8 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. A detailed list of possible threats is presented in (Wohlin et al., 2000).
**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 NWORDE have been collected using a computerised tool and are therefore reliable. The data collection for NCHANGE, NUC, NACTOR, and NREVISION involved human judgment. 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. Finally, because the study has been done in retrospect, there is a risk that the data is imprecise, e.g., that the available files (use cases models and use cases) are incomplete. The risk of imprecise data is present even when practitioners capture data when they work. For retrospective studies this risk is even higher. However, this threat is minimal, the records were complete, the version handling system had no dangling references to missing documents.

**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 (Loconsole and Börstler, 2003).

**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.

**Internal validity.** The study described here is a correlational study. We have shown that all measures are significantly correlated to number of requirements changes. 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 4.4. 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
6 Discussions and conclusions

In this paper, we have described a correlational study on number of requirements changes performed at BAE Systems Hägglunds AB, Sweden. We collected and analysed data of two historical projects in the company for five measures of size (NUC, NACTOR, NLINE, NWORD, NREVISION). Applying univariate and multivariate regression analysis, we built prediction models using data collected on a medium size software project. We then evaluated the models accuracy by applying the models on a set of data collected on a second, slightly larger, project developed at the same company. Our prediction models receive as input data for several measures of size for files describing software requirements. The models produce a number as output: the total number of changes to the requirements document. The dependent variable does not measure volatility directly. However, by applying the measure within specific time intervals, we can identify volatility trends. From an analysis of the volatility trends, 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.

Our hypothesis was: the size measures NACTOR, NUC, NWORD, NLINE, and NREVISION are good predictors of number of requirements changes. The data analysis showed that all measures have a positive significant relationship to NCHANGE. The measures are accurate predictors, except NUC and NREVISION that were associated to a low coefficient of determination. The accuracy of the measures as predictors is also shown by the low values of MMREs and high values of pred (see table 7). Looking at the coefficient of determination of the univariate models, the best predictor seems to be NLINE, followed by NWORD, NACTOR, NREVISION, and NUC. Similar results were obtained by measuring the MMREs (see table 7). The multivariate model 6, which combine the covariates NWORD and NREVISION, shows the best performance on data set A. It has the highest coefficient of determination, lowest MMRE, and highest Pred(0.25), but has weak performance on data set B. This may be due to the differences between the two data sets especially because NREVISION had different means in the two data sets (see table 2). When applying the models on data set B, NLINE was the best predictor, followed by NWORD, the multivariate model with covariates NWORD and NACTOR, and the univariate model NACTOR. The predictive ability of NLINE is comparable to those of known models such as COCOMO (Boehm, 1981). It is interesting to note that the best models are those which depend only on the length of the requirements documents and not on the functional-
ity or complexity. This result is in accordance with the principal component analysis, where the PC1, associated to the length, was the most influential.

Our models predict the number of changes to a certain requirements document by using different measures of size of requirements document as input. The problem of predicting number of requirements changes then, becomes a problem of estimating the size of a requirement document. How can we foresee that the size of a requirement will increase? If the size is small, the number of predicted changes will be small. Given a certain size, our models predict the number of changes for that size. When the size changes the models have to be applied again. Based on these findings, we suggest the following approach for using the prediction models:

1. Collect data on a current project (NUC, NACTOR, NLINE, NWORD, NREVISION, and NCHANGE).
2. Predict the number of changes in the current project using one of the models described earlier, for instance model 3. This particular model would receive values of NLINE as input and produce a predicted number of changes as output.
3. Compare the actual with the predicted number of changes. If the actual number is equal or higher than the predicted number of changes, there may be no more changes. Alternatively, the requirements document is an outlier and we need to control that document and eventually redesign it. If the actual number is less than the predicted number of changes, we have to expect more changes to that requirements document.

Our results suggest that variation in size of requirements specifications imply variation of number of changes. 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. Investigating many qualitative aspects on many requirements documents is expensive, subjective, and 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 models 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 measures NLINE and NWORD 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 upon 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 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 (MacDonell, 1997).

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.

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 NCHANGE was obtained by counting the number of changes to a unit of requirement document. The size of change is usually dependent on the effort spent for the change or the number of artifacts impacted by the change. Unfortunately, this information was not available in the current projects. Therefore, the size of change was estimated by the authors. 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 a word appears in red. We did not considered these words as changed.

References


N. Fenton and S.L. Pfleeger. Software Metrics: A rigorous and practical ap-


D. Zowghi and N. Nurmuliani. A study of the impact of requirements volatility on software project performance. In *APSEC ’02: Proceedings of the Ninth*
Annabella Loconsole obtained the Laurea degree in Computer Science at Bari University, Italy. She developed her Master Thesis at the Fraunhofer-IPSI, Darmstadt, Germany. She is currently a PhD student in software engineering at the Department of Computing Science, Umeå University, Sweden. Her main research topics are requirements management, software process improvement and empirical software engineering. She is a member of the IEEE Computer Society.

Jürgen Börstler is an associate professor and director of studies in computing science at Ume University, Umeå, Sweden. He received a Masters degree in Computer Science, with Economics as a subsidiary subject, from Saarland University, Saarbrücken, Germany and a PhD in Computer Science from Aachen University of Technology, Aachen, Germany. His main research interests are in the areas object-oriented software development, requirements management, process improvement, and educational issues in computer science.