An Investigation of Prediction Models for Project Management

D. Rodriguez, R. Harrison, M. Satpathy  
School of Cybernetics, Computer Science and Electronic Engineering  
The University of Reading  
Reading, RG6 6AY, UK  
Emails: {d.rodriguez-garcia, r.harrison, m.satpathy}@rdg.ac.uk

J. J. Dolado  
The University of the Basque Country  
20009, San Sebastian, Spain  
Email: dolado@si.ehu.es

Abstract
It has been claimed that dynamic prediction models can be used to help project managers make more accurate estimates than static prediction models. However, such a claim needs to be validated so that project managers can use dynamic models with confidence. In this paper, we discuss an experiment we conducted in an academic environment that compared a dynamic model using BBNs with a static model involving the COCOMO and Akiyama models. The results from this experiment in fact validate the above claim. However, we suggest replication of this experiment in order to increase confidence to our results.

Keywords: Software Estimation, Bayesian Belief Networks

1 Introduction
Project management is crucial to the success of any project. Proper management not only ensures that the project is completed in time using available resources but it also ensures that the product meets necessary quality requirements. Further, prediction of various attributes like cycle time, cost, effort, resource requirements, safety and reliability are of paramount importance to project managers. Advance knowledge of such attributes at intermediate stages during the project development not only provides the manager with the knowledge about the status of the project but also negative indicators warn of possible risks so that preventive measures could be initiated to minimize their impacts. In short, the project manager has greater control over the project.

Effective estimation of attributes such as the above not only requires a solid technical basis, but also knowledge of various parameters specific to the organization. However, in a realistic situation, much of the information about past projects which should help in the estimation of project parameters is unknown or uncertain. Moreover, the required data comes from multiple sources and the integration of all such data may be difficult without intelligent tool support. With this in mind, we have developed a project management tool called SEGESOFT [1].
SEGESOFT maintains various structures associated with projects such as Work Breakdown Structures (WBS), PERT charts etc. It also maintains Process Breakdown Structures (PBS) which document the decomposition of the whole development process into subprocesses, and the various quality attributes and metrics associated with them [2]. Further, the tool also maintains the quality attributes and corresponding metrics associated with important intermediate products (including the end product) generated during the development process. The SEGESOFT tool also incorporates simulation and dynamic modelling, knowledge discovery, and Bayesian Belief Networks, so that a manager can obtain and manipulate a broad spectrum of information to facilitate project management.

There are two main models that may be used for analysis and prediction of software quality attributes. Static models are usually represented by a set of equations that are based on experience and expert judgement. Examples include the COCOMO model by Boehm [3], and the Akiyama model [4], which predicts the number of defects in software in terms of the number of lines of code. It is to be noted that such equations were developed nearly twenty five years ago, whereas technology in the meantime has changed dramatically. Further, such static models do not take into account the causal relationships that exist between various quality attributes. For instance, the number of defects not only depends on testing and development efforts, but also on the experience of the developers and testers. Dynamic models alleviate such problems by taking care of the causal relationships between various attributes. Prominent dynamic models which are used in software engineering include system dynamics and Bayesian networks. In this paper we focus on Bayesian networks.

It is believed that dynamic models are superior to static models in general. However, such claims need to be validated and measurement is necessary to determine the extent of this superiority. Three types of validation techniques that are commonly employed in experimental software engineering are: surveys, case studies and experimentation [5, 6]. In this paper we will use experimentation.

Experimentation is usually done in a laboratory environment. Although it is very difficult to perform an experiment correctly [7], the objective is to manipulate one or more variables and control all other variables at fixed levels. The effect of the manipulation is measured, and statistical analysis is performed over the measured values to validate the initial assumptions. Experimentation can also be used to compare the effectiveness of two different methods. In this paper, we use experimentation to find out if dynamic models are superior to static models in relation to prediction and analysis.

The organization of the paper is as follows. Section 2 introduces our static and dynamic models. Section 3 discusses our experimental set up. Section 4 discusses the execution of the experiment. In section 5, we describe a statistical analysis of the experiment data. Section 6 concludes the paper.

2 Static and Dynamic Models

2.1 The Static Model

We use a combination of the COCOMO model [3] and the Akiyama model [4] as our static model. We chose these because of their simplicity. The COCOMO model predicts total
effort that would be necessary to generate a source program of certain complexity. The prediction is represented by the equation: \( \text{eff} = c \cdot L^b \), where \( \text{eff} \) is the effort in man months, \( L \) is the code size in KLOC, \( c \) and \( b \) are constants, dependent on the complexity of the problem (organic, semi-detached or embedded).

The model by Akiyama presents a relationship between the number of defects discovered in a source program and the size of code: \( \text{def} = 4.86 + 0.018 \times \text{LinesOfCode} \), where \( \text{def} \) is the number of defects introduced (i.e. the sum of the number of defects found during testing and those discovered within 2 months of delivery).

In the present experiment, we use function points (FP) to represent functionality [8]. We also assume that a function point, on average is translated into 53 lines of C++ or Java code[9]. For example, to use these formulae in our experiment to predict the number of defects In a system with 100 FP assuming that the complexity of the system is organic: 100 FPs results in 5.3 KLOC and so COCOMO predicts \( \text{eff} = 2.4 \times (5.3)^{1.05} \sim 13.8 \text{ man-month} \), and so the Akiyama model predicts that \( \text{def} = 4.86 + 0.018 \times 5300 \sim 100 \text{ defects} \).

### 2.2 BBN as a Probabilistic Dynamic Model

A Bayesian Belief Network (BBN) [10, 11] is a directed graph in which the nodes represent uncertain variables and the arcs represent the causal relationship between the variables. Each node has a probability table, which stores the conditional probabilities for each possible state of the variable in relation to each combination of its parent state values. For a node without any parent, such a table stores the marginal probabilities for each possible state of that node. If the state of a certain node is known then its probability table is altered to reflect this knowledge. Such knowledge is then propagated to determine the changed probabilities of (of all possible values associated with) other nodes. Note that the initial probabilities of the nodes in a BBN are obtained from expert judgment and past project data. In fact, tools are available to help in the generation of BBNs from historical project data [12]. They have been used in various application areas ranging from medical diagnosis to software engineering.

Since the conditional probabilities of the probability tables associated with the nodes of a BBN are determined with respect to past project data and expert judgment, it is expected that they represent the approximate causal relationships with respect to the various quality factors and attributes (identified by the nodes) of the organization concerned. So, when the knowledge about certain factors (i.e. nodes of the BBN) are known, the probability tables can be used to effectively predict the values associated with other nodes in the network. Note that when an organization evolves, the effect of this evolution can be incorporated into the BBNs by modifying the probability tables accordingly.

BBNs can serve as decision support systems when working with uncertainty. In software engineering, it is almost impossible to predict exact values for quality and estimations; in fact, it is usually sufficient to deal with ranges or intervals of parameters. BBNs allow us to represent intervals indicating values to which the parameter must belong. Also their visual support helps in understanding the causal effects.
3 Experiment Set-up

3.1 Goal of the Experiment

We use a goal definition template [13] to state the objectives for our experiment. This template has five sub-headings which we used as shown below:

- **Object of study:** BBN as a dynamic model and equations as a static model
- **Purpose:** To compare the effectiveness of BBN verses the static equations
- **Quality Focus:** estimation capability of both the models
- **Perspective:** from the viewpoint of project manager/ researcher
- **Context:** This experiment is conducted in an academic environment with post graduate students and researchers as subjects. This experiment is conducted as a blocked subject-object study [14].

The BBN that we used for our experiment is shown in Figure 1. This BBN represents a defect estimation model. Each node of the graph represents a variable, and an arrow from node $m$ to node $n$ represents a causal relationship between variable $m$ to variable $n$. Each variable can take one of its allowable values; for instance, the variable *functionality* can take one of the 7 permitted values (Figure 2). In the BBN of Figure 1, the variable code size depends on the *functionality* and *complexity* of the software. *Defects introduced* (defects in source code) depends on the *code size* and the *design effort*. *Residual defects* (defects that remain after delivery) depends on the defects in the source code and the defects detected during testing. The number of defects detected during testing depends on the amount of testing effort and the number of defects in the source code. Dependencies are characterized by a probability table. For instance, if an organic project with 600 FP has average testing effort and design effort then the BBN infers that the number of defects introduced in the source code will be between 500 and 600. Figure 3 demonstrates such a scenario.
The usage of the tool incorporating this BBN is fairly intuitive. The tool has two modes of operation: observation mode which allows insertion of evidence, and a query mode where estimations are visualised. Black nodes in the network mean that evidence has been entered, and grey ones mean that no evidence has been entered and we can query their probabilities under the impact of the supplied evidence.
3.2 Hypotheses

The null hypotheses are as follows:

- \( H_{0,1} \): There is no difference between the estimates predicted by the static model versus the same predicted by the BBN.
- \( H_{0,2} \): There is no timing difference between estimations predicted by BBNs with tool support and the same predicted by the static model without tool support.

The alternative hypotheses that we expect to prove with this experiment can be defined as follows:

- A BBN makes the context of the problem under study clearer than the static model does.
- BBNs make it easier to answer a broad spectrum of questions relating to software concepts and estimates than static models.
- There are some specific management decisions which can be easily calculated by a BBN through backward propagation of probabilities, whereas such questions cannot be answered using a static model, e.g., how resources can be used to produce a product with a given defect density.

3.3 Subjects

The subjects of the experiment were postgraduate students and researchers at the University of Reading. They all had very similar educational qualifications in the field of computer science. Following the terminology of Wohlin et al. [14], we have adopted the approach of *convenience sampling* as regards to the selection of our subjects.

We allocated each subject exactly one treatment, i.e. either the use of BBN or the use of static equations. The experiment was conducted on an individual basis because it was impossible to carry out the experiment with all the subjects simultaneously. In order to minimize threats to the experiment the same documentation was given to all subjects prior to the start and then the subjects were assigned to a pre-determined model. We assigned an equal number of subjects to each category of treatment.

3.4 Experimental Variables

In our experiment, the independent variables were:

- BBN model representing the defects estimation problem
- Static equations represented by the Akiyama and Boehm models.
The dependent variables were:

- **interest** (S1) of the subjects in the area under study (we measure the degrees of agreement using a five point Likert scale where 1 is fully agree and 5 fully disagree)
- background **knowledge** (S2) of the subjects in the area under study (1 for each correct answer and 0 for each incorrect one)
- the **subject-score** (S3), the score the subjects obtain from their questionnaires (1 for each correct answer and 0 for each incorrect one).
- the **subject-time** (S4), the time that subjects take to complete Section 3 of the questionnaire (in minutes).

In this paper our analysis is mainly based on the **subject-score** and **subject-time** variables. Although further analysis will be performed with the **interest** and **knowledge** variables, we used them mainly to check if the samples in both groups were similar.

### 3.5 Experiment Procedure

We prepared a questionnaire with four sections. The first section contains questions about the subject’s interest in the field of Software Engineering. The questions in Section 2 focus on the subject’s knowledge in software testing. Section 3 contains questions which involves the use of a model (static or dynamic) to compute and answer the values of some attributes relating to testing. Section 4 has questions relating to the subject’s impression on and interest in the approach. Before starting the experiment, each subject was given 10 minutes of introduction to the experiment. They were then asked to read and make sure they understood the given documentation. They were also allowed to ask questions and clarify doubts before answering the questionnaire. Subjects using the dynamic model were told how to use the tool support which had the Bayesian network encoded in it.

The expected answers to Section 3 were calculated using (i) the static model (equations) and (ii) the dynamic model (BBN). In this way we could guarantee that the expected answers for both the static and dynamic groups were in fact the same.

### 4 Experiment Results

We wanted to ascertain whether the use of a dynamic model using a BBN improved the estimation of various attribute values, and so we will subject our hypotheses to one-tailed analysis. Data was collected from 12 subjects, half of them used the BBN tool and the other half calculated the equations with spreadsheets or calculators. Tables 1 and 2 show the raw data collected.
### Table 1 Scores of the subjects who used BBN

<table>
<thead>
<tr>
<th>Subjects (BBN)</th>
<th>Total score</th>
<th>Total Score</th>
<th>Total Score</th>
<th>Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Section1(S1)</td>
<td>Section2(S2)</td>
<td>Section3(S3)</td>
<td>Section3</td>
</tr>
<tr>
<td>B1</td>
<td>13</td>
<td>3</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>B2</td>
<td>11</td>
<td>4</td>
<td>10</td>
<td>21</td>
</tr>
<tr>
<td>B3</td>
<td>13</td>
<td>2</td>
<td>8</td>
<td>21</td>
</tr>
<tr>
<td>B4</td>
<td>14</td>
<td>1</td>
<td>5</td>
<td>31</td>
</tr>
<tr>
<td>B5</td>
<td>12</td>
<td>4</td>
<td>8</td>
<td>22</td>
</tr>
<tr>
<td>B6</td>
<td>14</td>
<td>2</td>
<td>7</td>
<td>24</td>
</tr>
<tr>
<td>Mean</td>
<td>12.83</td>
<td>2.67</td>
<td>7.67</td>
<td>22.5</td>
</tr>
<tr>
<td>Mode</td>
<td>13</td>
<td>4</td>
<td>8</td>
<td>21</td>
</tr>
<tr>
<td>STD</td>
<td>1.17</td>
<td>1.21</td>
<td>1.63</td>
<td>4.93</td>
</tr>
<tr>
<td>VAR</td>
<td>1.37</td>
<td>1.47</td>
<td>2.67</td>
<td>24.3</td>
</tr>
</tbody>
</table>

### Table 2 Scores of the subjects who used the static model

<table>
<thead>
<tr>
<th>Subjects (Static)</th>
<th>Total score</th>
<th>Total Score</th>
<th>Total Score</th>
<th>Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Section1(S1)</td>
<td>Section2(S2)</td>
<td>Section3(S3)</td>
<td>Section3</td>
</tr>
<tr>
<td>St1</td>
<td>11</td>
<td>4</td>
<td>10</td>
<td>57</td>
</tr>
<tr>
<td>St2</td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>46</td>
</tr>
<tr>
<td>St3</td>
<td>11</td>
<td>2</td>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td>St4</td>
<td>15</td>
<td>3</td>
<td>8</td>
<td>26</td>
</tr>
<tr>
<td>St5</td>
<td>11</td>
<td>1</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>St6</td>
<td>13</td>
<td>4</td>
<td>5</td>
<td>39</td>
</tr>
<tr>
<td>Mean</td>
<td>11.5</td>
<td>2.83</td>
<td>5.33</td>
<td>38.67</td>
</tr>
<tr>
<td>Mode</td>
<td>11</td>
<td>4</td>
<td>#N/A</td>
<td>#N/A</td>
</tr>
<tr>
<td>STD</td>
<td>2.35</td>
<td>1.17</td>
<td>3.08</td>
<td>12.39</td>
</tr>
<tr>
<td>VAR</td>
<td>5.5</td>
<td>1.37</td>
<td>9.47</td>
<td>153.47</td>
</tr>
</tbody>
</table>

Figures 4 to 7 show boxplots of the dependent variables. From the plots in Figures 4 and 5, it can be seen that the groups had very similar interest and background in the area under concern (they have similar median values).
On the other hand, from the boxplots of Figures 6 and 7, it can be inferred that on average, subjects in the BBN group scored better than the other group, and further, they also took less time. It can be said that the group using tool support showed improved performance in terms of time compared to the other group. However, we believe that the static group needed much more time in order to decide which equations to apply.

4.1 Dependent Variables: subject-score (S3)

Since we wanted to analyse the hypotheses involving two treatments, we applied the independent t-test to investigate the effect of the independent variables on the dependent variables subject-score and subject-time under both treatments. As the sample is small, we decided to use an alpha value of 0.1 (\( \alpha = 0.1 \)). Therefore the confidence of all decisions to reject or accept the hypotheses \( H_{0,1} \) and \( H_{0,2} \) is 90%.
Table 3 shows *subject-score* variable group statistics for Section 3 of the questionnaire for both treatments. Table 3 shows that, for the variable *subject-score*, the mean is 7.67 for the group using BBNs, compared with 5.33 for the group using static equations.

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>bayesian</td>
<td>6</td>
<td>7.67</td>
<td>1.63</td>
<td>0.67</td>
</tr>
<tr>
<td>static</td>
<td>6</td>
<td>5.33</td>
<td>3.08</td>
<td>1.26</td>
</tr>
</tbody>
</table>

**Table 3 Group statistics for the Student-Score variable**

Levene’s Test checks to define whether the variances are homogenous [15]. Table 5 shows that the value of p = 0.111 > 0.05, and therefore, we can apply the t-test assuming that the variances are equal. The rest of the columns in Table 4 show the different parameters of the t-test. It is possible to reject $H_{0,1}$ because the probability is below the $\alpha = 0.1$ considered. However, we need to accept this result with caution since both the t-values are relatively close.

<table>
<thead>
<tr>
<th>S3</th>
<th>Levene’s Test</th>
<th>t-test for Equality of Means</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F</td>
<td>Sig.</td>
</tr>
<tr>
<td></td>
<td>Equal variances assumed</td>
<td>3.06</td>
</tr>
<tr>
<td></td>
<td>Equal variances not assumed</td>
<td>1.64</td>
</tr>
</tbody>
</table>

**Table 4 Independent t-test for Equality of Means (Subject-score variable)**

It is worth mentioning that one of the questions (Question 3.6 in the Appendix) asked the subject to estimate design effort for a given defect density. Such a question is difficult to answer using a static model since the defect density depends on many variables and the influence of each variable on the outcome can not be known. In our experiment, only two of the subjects using the static model answered the question correctly. However, subjects using BBN modelling could answer the question easily because of the backward propagation of probabilities in the BBN.

### 4.2 Dependent Variables: *subject-time* (TIME)

We also performed a one tailed t-test with an alpha value of 0.1 ($\alpha = 0.1$) with the variable *subject-time*. Table 5 shows the group statistics of the timing measures for Section 3 of the questionnaire for both the treatments. Observe that the means are significantly different.

<table>
<thead>
<tr>
<th>TIME</th>
<th>N</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>bayesian</td>
<td>6</td>
<td>22.5</td>
<td>4.93</td>
<td>2.01</td>
</tr>
<tr>
<td>static</td>
<td>6</td>
<td>38.67</td>
<td>12.38</td>
<td>5.05</td>
</tr>
</tbody>
</table>

**Table 5. Group Statistics for the time variable**

Since the p-value of Lavele’s test is 0.11 (> 0.05), we can assume that the condition for equal variance holds for the t-test. We can clearly reject the null hypothesis $H_{0.2}$ as the p-value is smaller than the proposed $\alpha = 0.1$. 
4.3 Power Analysis and Sample Size

Power analysis is directed towards exploring the different situations that could arise with respect to the effect size, significance level and power level [16]. This exploration, jointly with the post-analysis of the actual results will help us to formulate future similar experiments.

Alpha, the significance level, is the probability of committing a Type I error, i.e., to reject incorrectly the null hypothesis, when it is true. In our case, this will tend to promote, incorrectly, one of the methods (the BBN approach). However, we assume that this will not have an adverse effect on the estimations. On the other side, committing a Type II error, that is, to accept incorrectly the null hypothesis when it is false, would imply that we will be losing the benefits of one of the methods (the BBN method).

In the current setting we did not have previous values of the effect size, so we depicted the combinations allowed, by assuming an error standard deviation of 0.5 (and 1), alpha of 0.1 and the fixed sample size of 12 subjects. This kind of exploration is the last resort we have when no other similar studies are available.

In Figure 8, we have an exploration of one of the possible situations and the power curve obtained (with the software JMP [17]). In Figure 9, we depict another curve, by assuming another error standard deviation of 1. We observe that a power level above of 0.5 is obtained if the difference detected between the means is above 0.64. If the assumed error standard deviation is 1 we have a power level above 0.5, if the difference detected is above 1.03.

The final differences observed in TIME are above 1. Therefore, given the results of Figures 8 and 9, we can be reasonably confident about the conclusions of the tests of significance.
5 Threats to the Experiment

A pilot study was carried out with colleagues in the Applied Software Engineering Research Group of the University of Reading. This pilot study helped to improve the questionnaire and the associated documentation.

5.1 Conclusion Validity

Conclusion validity is concerned with the relationship between the treatment and the outcome. One factor affecting the experiment could be the small number of subjects. This is known to have a negative effect on the power of the statistical methods reducing the chance of finding an effect if it exists.

5.2 Internal Validity

Internal validity is concerned with the relationship between the treatment and the outcome; i.e., if the conclusions can be obtained from the causal effect of the independent variable. A crucial step in the experimental design consists of minimising the impact of the threats to the validity; i.e., minimising factors that can affect the dependent variables without the researcher’s knowledge.

The volunteers for our experiment were chosen from a group of post graduate students and researchers in our department. This may explain why no subject took the experiment lightly and also why no subjects dropped out of the experiment. In addition, the subjects were randomly assigned to one of the treatments in order to avoid selection effects.

There could be some maturation effect due to learning and practice as the experiment proceeded. Some of the subjects in both the groups modified some of their initial answers as they became familiar with the tool or with the equations in the static model (some of the
subjects wanted to understand the questions better). There is also a risk that the people using the static model needed more training time to master how to apply the formulae correctly.

5.3 Construct Validity

Construct validity is concerned with the degree to which the variables used in the study accurately match the concepts they intend to measure. Our BBN model may not adequately capture the advantages of using BBNs in a general software engineering scenario, since our experiment took a simplistic view of the problem and the specific advantages or disadvantages of using BBNs can not be captured by the experiment.

The BBN and the static models were created in such a way that it would be possible to answer the questionnaire using both models. However, it was difficult to create fair scenarios to compare both models. To deal with this threat, the static model was quite simple and the BBN simulated the static model, so the ‘correct’ answers in the questionnaire were same for both groups.

In order to gauge the background knowledge of the subjects in software engineering in general and on testing in particular, we referred to the Software Defect Reduction Top 10 List [18]. After analysing the results of the experiment, we believe that more general questions could have given more accurate results. It is also very difficult to measure interest objectively.

5.4 External Validity

External validity is concerned with the degree to which the results of the research can be generalized.

Although the subjects had experience in developing software, most of them had no industrial experience. Therefore, it is difficult to generalise the results. Further studies should be carried out to assess the usefulness of BBNs as used by experienced project managers.

As the experiment was carried out on an individual basis, we believe that no misunderstanding over the questions occurred and all of the subjects took the experiment seriously, but we also believe that some of the subjects may have tried to guess some of the answers.

6 Conclusions and Future Work

For a probabilistic approach to project management, BBNs can be useful in the sense that they can explain some cause-effect relationships of software engineering much better than static models. Also, the backwards propagation of the probabilities in the Bayesian Networks can help in managerial decisions that might not be possible using static models. In order to validate such a statement we carried out an experiment that compared a probabilistic dynamic model with the classical COCOMO and Akiyama models. We measured two metrics: subject-score, the number of correct answers in a section of the
questionnaire, and the total time to complete this section. Our results showed statistically positive results in favour of the group using the probabilistic model. However, since the statistically significant test to reject the null hypothesis in relation to the subject-score is close to the limit of acceptance, further empirical studies needs to be performed.

We have learnt some important lessons from this experiment, and this experience will feed into our replication of this experiment. For instance, Section 2 had some ambiguous questions which must be avoided and more generic questions will be used to gauge the subjects knowledge in software engineering. Although we made sure that the subjects understood how to use the equations of the static model, we observed that, at times, their interpretation was not straightforward.

Our future work will be to overcome some of the difficulties we found while using BBNs in software engineering. There should be a systematic approach to creating and modifying the BBN probability tables. However, we believe that BBNs could be made more adaptable to a particular environment than the classical static models. In addition, better multimedia Graphical User Interfaces with access to BBN integrated with project management environments are necessary. Further experiments will consider real project data to investigate the advantages of BBNs over the traditional regression models. Also, it would be of interest to compare BBN approach with other estimation techniques such as estimation by analogy [19].

Acknowledgments

This work has been supported by The University of Reading, UK. Also this work is based on a project management tool developed by the SEGESOFT Group, supported by CICYT TIC2001 1143-C03-01 (Spain). The authors thank postgraduates and research staff for carrying out this experiment and the members of the Applied Software Engineering group at The University of Reading for their useful comments.

7 References


Appendix - Questionnaire

Section 1. Interest in software engineering (5 min)
Please, tick one of boxes attached to each question according to your opinion in accordance with the following criterion.

1 2 3 4 5
agree ○ ○ ○ ○ ○ disagree, where the meaning of each box is as follows:
1 fully agree
2 agree
3 undecided
4 disagree
5 fully disagree

S1.1. I consider it very important for computer science students to know as much as possible about the principles of software engineering.

1 2 3 4 5
agree ○ ○ ○ ○ ○ disagree

S1.2. I consider that the knowledge of project management is essential to coordinate the various software development activities and to ensure good quality of the final product.

1 2 3 4 5
agree ○ ○ ○ ○ ○ disagree

S1.3. I consider that quality of a software product can be achieved by applying sound software engineering principles.

1 2 3 4 5
agree ○ ○ ○ ○ ○ disagree

S1.4. I do NOT consider software development as an art and I think that the only way to achieve product or process quality is to apply software engineering principles.

1 2 3 4 5
agree ○ ○ ○ ○ ○ disagree

S1.5. I agree that the number of defects found in a software after delivery is a primary indicator of customer satisfaction.

1 2 3 4 5
agree ○ ○ ○ ○ ○ disagree

S1.6. I consider that testing is a very important software engineering phase in the development process and it will influence the customer’s point of view about the quality of the product.
Section 2. Knowledge about Software Engineering (5min)

For each question tick exactly one answer. If you consider that several options are correct, choose the one that you feel is most appropriate. This section will be used to gauge your previous knowledge in the area.

S2.1. For a typical software project, finding and fixing a software problem after delivery is about

- ○ 3 times
- ○ 5 times
- ○ 10 times
- ○ 100 times
- ○ 125 times

more expensive than finding and fixing it during the requirements and early design phases.

S2.2. About X percent of defects come from Y percent of the modules.

Where X: □ 10 % □ 20 % □ 50 % □ 80% □ 90%  
and Y: □ 10 % □ 20 % □ 50 % □ 80% □ 90%

S2.3. About X % of the downtime comes from, at most, Y percent of the defects.

Where X: □ 10 % □ 20 % □ 50 % □ 80% □ 90%  
and Y: □ 10 % □ 20 % □ 50 % □ 80% □ 90%

S2.4. Peer review (inspections) catch X% of the defects.

where X:  
- ○ 10 %
- ○ 20 %
- ○ 50 %
- ○ 60%
- ○ 80%

S2.5. Disciplined personal practices can reduce defect introduction rates by up to X percent.

Where X:  
- ○ 10%
- ○ 25 %
- ○ 50 %
S2.6. The number of pre-release defects in an implementation is primarily a function of:
- code size (KLOC)
- problem complexity
- experience of the developers.
- number of program modules
- maturity of the organization

S2.7. The effort spent on testing is around what percent of the total effort spent on a software project for an average size project (20KLOC).
- 0-10%
- 10-20%
- 20-30%
- 30-40%
- 40+

Section 3 Questions about project management focusing on product quality

Please, note down the time when you start this section.
TIME: ________ hours ________ min.

S3.1. Assume a project of 600 function points (FP). Also assume the project is organic, average testing effort and average design effort. What will be the number of defects introduced (no. of defects after implementation)?
- 0 – 100
- 100 – 200
- 200 – 400
- 400 – 750
- 750 – 100
- 1000+

S3.2. For the same project (in S3.1), will you achieve less than 100 residual defects (post-delivery defects) ?
- Yes, I will achieve that level of quality
- No, unless I increase testing effort and/or design effort
- No, I cannot achieve that level no matter how much I increase both testing effort and design effort.

S3.3. Assume a project has been estimated to have 200 function points (FP). The software is to be implemented in C++ and it is an embedded system. What is the estimated total effort for the project in man-month?
S3.4. Assume an organic project. You are in charge of developing a new system of 350 *function points* (FP). Your client specified in the contract a maximum of 100 *residual defects* (post delivery). Your manager has a lot of projects at the moment so (s)he doesn’t allow you more than average resources for *testing effort* and *design effort*. Would you agree to lead that project?

- Yes, it is possible with current resources.
- No, unless I increase the design effort from average to high.
- No, unless I increase the testing effort from average to high.
- No, unless I increase both efforts (design and testing)
- No, I cannot do it.

S3.5. And the same project (in S3.4) but with the size of 450 FP?

- Yes, it is possible with current resources.
- No, unless I increase the design effort from average to high.
- No, unless I increase the testing effort from average to high.
- No, unless I increase both efforts (design and testing)
- No, I cannot do it.

S3.6. Assume an *organic* project and average *testing effort*. A system of 600 *function points* FP has shown around 500 *residual defects* (post-delivery defects), what can you tell about its *design effort*?

- Design Effort very high
- Design Effort high
- Design Effort average
- Design Effort low
- Design Effort very low

S3.7. A module of 30KLOC has shown very low *defects detected* (pre-release faults), close to 100 but many *residual defects* (post-release faults), more than 800, state which one is the most probable cause?

- Both design effort and testing effort very low
- Design effort very low and testing effort low
- Design effort low and testing effort average
- High Complexity
- None of the above
S3.8. Checking the given documentation can you enumerate what variable(s)/factor(s) affect the number of defects introduced?
____________________________________________________________________
____________________________________________________________________

S3.9. Checking your given documentation, can you enumerate what variable(s)/factor(s) affect the total effort?
____________________________________________________________________
____________________________________________________________________

S3.10. A client was complaining about a component of 10 KLOC which has shown a high number of defects once installed in the client site (let’s say more than 200 defects after delivery –residual defects–). The same system was installed in another client but you have not received any complain yet. Could you give a logical reason for this?
____________________________________________________________________

Please, note down the time when you finish this section.
TIME: _____hours ______ min.

Section 4. Disturbing/Influence factors and comments (5 min)

S4.1. Are you interested in getting the results of the experiment and the solutions for the questionnaires?
○ Yes, I’m interested. My e-mail address is: _______________________
○ No, I'm not interested

S4.2. Please, tick if you did NOT have enough time to
○ read the materials on provided with the questionnaire
○ familiarize and understand the tool(s)
○ complete the test

S4.3. How much of confidence do you have in answering the questions of section 3?
○ I’m sure I answered the questions correctly
○ Quite certain I answered most of the questions correctly
○ Not sure, I got a bit lost
○ I didn’t care, so I don’t think I did well.

If you have used the tool, please answer the following questions

S4.4. Have you ever used an estimation tool (or a project management tool) while developing your projects?
○ Yes
S4.5. Do you consider it appropriate to link the techniques like BBN with project management tools?

- Yes
- No
- Not sure

S4.6. With the current limitations of the prototype, do you consider that this approach can be useful to project managers?

- Yes
- No
- Not sure

S4.7. What is your impression of the tool?

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

S4.8. How do you think this tool or approach can be improved?

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

S4.9. Would you like to add something else?

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________