mgr Łukasz Radliński[†‡] prof. Norman Fenton[‡] dr Martin Neil[‡] dr David Marquez[‡]
[†]Instytut Informatyki w Zarządzaniu, Wydział Nauk Ekonomicznych i Zarządzania, Uniwersytet Szczeciński
[‡]Department of Computer Science, Faculty of Informatics and Mathematical Sciences, Queen Mary, University of London, United Kingdom

Modelling Prior Productivity and Defect Rates in a Causal Model for Software Project Risk Assessment

Abstract

Estimating effort and quality of developed software has been a demanding task for project managers. Many models based on different approaches have been proposed to solve this problem. Most of them dealt only with one of either estimating effort or quality. Our aim is to develop a causal model capturing both of these and enabling trade-off analysis between the functionality, effort and quality. This model incorporates several quantitative and qualitative factors influencing the software development process and product. One of the key challenges is to be able to use some prior knowledge about the productivity and defect rates in the model. This knowledge can be obtained from the literature, a company's data about past projects or assessed by an appropriate software project management expert. In this paper we present the results of an analysis which we performed using mainly ISBSG database of software projects. We also compared them with analyses available in the literature. These results are incorporated in our model for more accurate predictions especially when software companies do not have appropriate data about their past projects.

Causal model for software project risk assessment

Much effort has been spent in developing models for estimating software size, development effort and delivered quality. These models rarely attempted to capture all these variables. Our ultimate aim is to include them in a single causal model together with other factors affecting them.

Figure 1 illustrates schematic view of the productivity model [10] which extends the Bayesian net (BN) models developed in the MODIST project [9]. The models in MODIST had to be extended because empirical data about productivity and defect rates (PDR) were 'hard coded' into the model in such a way that it was difficult to use any new, more relevant prior data (the 'priors' were effectively biasing the model too much). The key part of the new model is the trade-off component (at the bottom of the figure) between:

- functionality – expressed in number of software units delivered,

- effort – expressed in both project duration and number of people, which are then adjusted by a 'Brooks factor' [1],
- quality – expressed as number of defects.

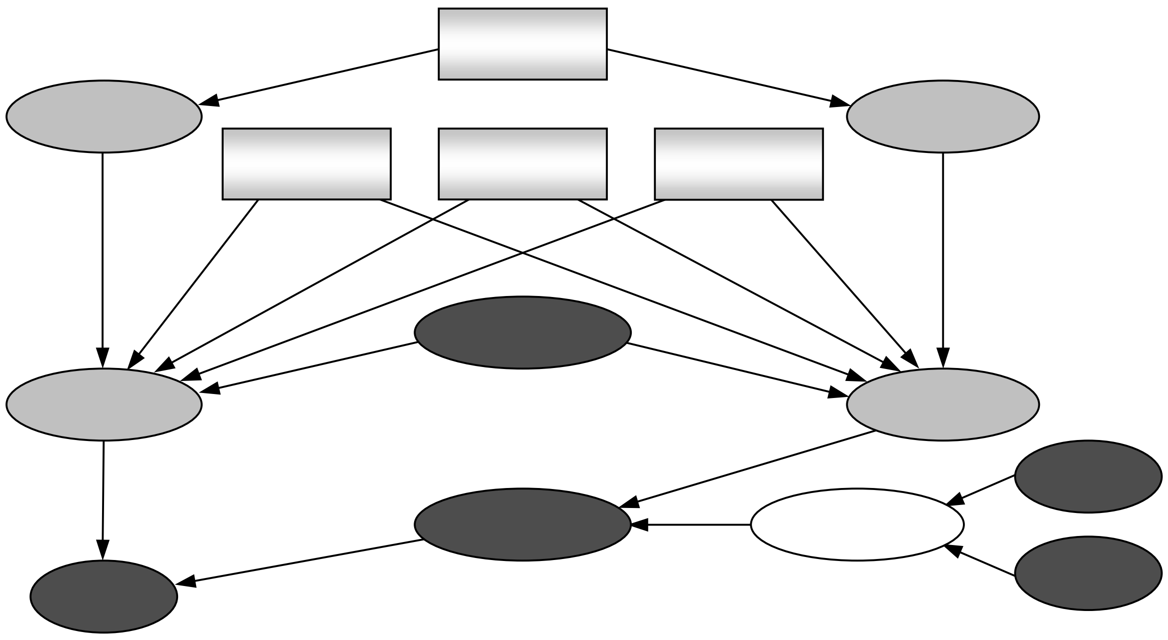


Figure 1. Schematic view of the productivity model

This trade-off component enables us to analyse relationships between the variables. For example, it can answer questions like: 'what effort will we need do develop software of some specific functionality (size) given that it must not exceed more than some specific number of defects?'.

These relationships are influenced by productivity and defect rates which are estimated by the model and which depend on inherent project factors (complexity, novelty etc.), requirements quality (completeness, stability etc.), and process and people quality (staff experience, following defined processes or CMM level etc.). A core component of the model is the idea that all project effort is devoted either to writing new code (which determines the productivity as defined by code delivered) or to other activities which are lumped together as 'quality assurance'. The greater the percentage of effort spent on 'quality' (getting the code right) the less effort is available for producing new code. Hence we have the node 'percentage difference of effort devoted to quality' and prior defect and productivity rates. Except for the prior rates, these factors reflect the difference between the current project compared with a 'typical' past project. The prior rates reflect their values for a 'typical' past project. They can be either:

- left with default values (probability distributions) – this inevitably increases the uncertainty (variance) of the resulting predictions;
- entered into the model as observations or soft evidence – if users can estimate them outside the model;
- estimated by the model – if users can provide some information about the past projects.

In this paper we focus on the latter scenario. We analyze factors influencing prior productivity and defect rates. Then we demonstrate how the results of this analysis are incorporated in our model.

Factors influencing productivity and defect rates

The ISBSG database [3] contains data about 3024 software projects described by 100 variables. For data analysis we selected only those variables which were identified as important in estimating PDR in relevant literature [2, 5, 6, 7, 8, 11, 12]. We also reduced the dataset by removing outliers and filtering data in the following stages:

- We removed projects with productivity rate outside the 5th to 95th percentile. Because the 5th percentile equalled to the lowest value for defect rate we kept all those projects for the analysis and removed only those above 95th percentile.
- We filtered data by leaving only those projects for which: data quality was classified as ‘A’ or ‘B’ (meaning good quality); function points count approach was ‘IFPUG’; and resource level included only development team effort as suggested in [4].

As a result we had data about 260 projects for analyzing defect rate and 1880 for productivity rate. For these empirical defect and productivity rates we fitted probability distributions. Among the several probability distributions analysed, the Log Normal appeared to best fit these empirical data for both productivity and defect rates, either for the whole dataset used in the analysis or by grouping projects using different categories of descriptive data. However, for some projects no defects were found and so the defect rate was 0. The Log Normal distribution accepts only input data with values above 0, but we wanted to avoid excluding them from the analysis. Hence, we adjusted the value of ‘number of defects’ by entering very small positive value (0.1) for projects which actually had no defects. As a result we achieved a small positive value of defect rate for these projects, which we could transform using the Ln function and so keep these projects in the analysis.

We also analyzed correlations between potential predictor variables suggested in the literature and PDR. Our analysis confirmed that most of these predictor variables were correlated (linearly or not linearly) with both productivity and defect rates. However, a few of them influenced only one of productivity or defect rate. For predictor variables expressed on a nominal scale we performed analysis of variance (ANOVA). This also confirmed that most of these predictor variables influenced PDR. We have used results from ANOVA in the expressions for productivity and defect rates adjusted by specific predictor variables.

Bayesian Net (BN) for prior productivity and defect rate estimation

Currently this model estimates productivity rate in 'number of function points delivered per person-hour' and defect rate in 'number of defects per function point'. We used the native units of measurement for software size and effort because of their popularity in practice and to avoid transformation inaccuracies from function points to e.g. KLOC (thousands lines of code) or person-hours to e.g. person-months.

We have analyzed several structures of the net to be applied for this model. The structure that best captures all necessary relationships is a structure where predicted variables (PDR) depend on a set predictor variables. Here predicted variables are expressed as functions of predictor variables. This would cause large Node Probability Tables (NPTs) for predicted variables. We would have to analyze every possible combination of states of predictor variables. We were not able to do it because of the lack of necessary volume of project data – the majority of combinations did not contain even a single observation and only for very few combinations would we have had enough observations to be able to build an inference rule (expression for predicted variable).

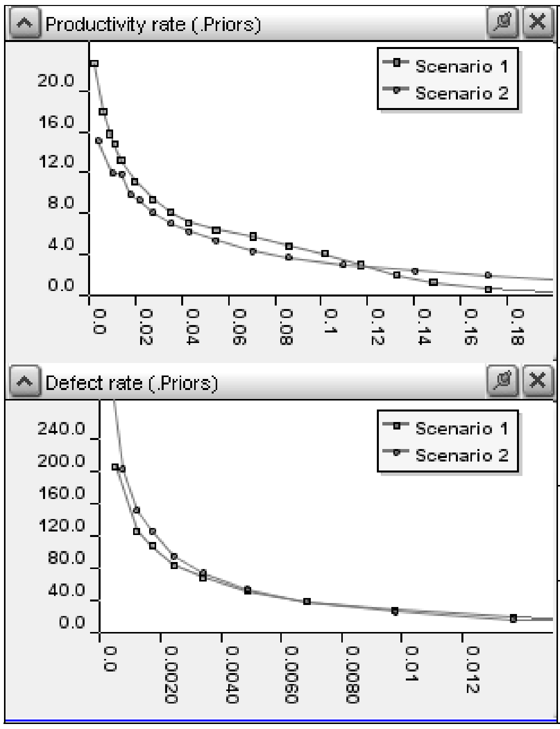
In our BN we decided on a different structure. For each predictor variable the model calculates the adjusted productivity or defect rate. These adjusted rates are Log Normal distributions estimated mainly from the ISBSG data. For example, 'productivity rate adjusted by language type' has the following partitioned expressions defined:

- for language type = '2GL': Log Normal($\mu = -2.46498$, $\sigma = 0.874257$),
- for language type = '3GL': Log Normal($\mu = -2.43887$, $\sigma = 0.813331$),
- for language type = '4GL': Log Normal($\mu = -2.04536$, $\sigma = 0.781440$),
- for language type = 'Application generator': Log Normal($\mu = -1.83327$, $\sigma = 0.565044$).

For some of the descriptive (labelled) predictor variables in the BN we had to limit number of states because of lack of necessary volume of data. For example, in variable 'Organisation type' we only allowed observations for states for which we had more than 5 projects' data. But we still had to face the problem of missing data. We included 30 states for 'Organisation type'. So we had to define 30 expressions for estimating 'productivity rate adjusted by organisation type'. That was not a problem. On the other hand, we also had to use these states for estimating 'defect rate adjusted by organisation type'. But only for a few states we could do it using ISBSG dataset because of the low volume of data. For the rest we entered the probability distribution estimated for the whole dataset (without categorization).

The final PDRs are weighted geometrical means of these adjusted rates. The weights are combinations of correlation strength estimated from ISBSG data, other additional data from relevant literature and our subjective belief on the strength of impact of predictor variable on PDR.

The BN also included variables which do not exist in the ISBSG database but which are known to have significant impact on PDR, such as CMM level [2].



	Scenario 1	Scenario 2
Observations:		
Application type	Decision Support System	Stock control & order processing
Architecture	Client server	Multi-tier with web public interface
CASE tools used	No	Yes
CMM level	1	4
Development platform	Multi	Mid range
Development type	New	Enhancement
Is Web	No	Yes
Methodology acquired		Traditional
Methodology used	No	Yes
Organisation type	Banking	Wholsale & retail trade
Prog. language type	3GL	4GL
Predictions: productivity rate	Mean: 0.05270 Median: 0.03995 SD: 0.05994	Mean: 0.08255 Median: 0.05582 SD: 0.10666
Predictions: defect rate	Mean: 0.00881 Median: 0.00463 SD: 0.02403	Mean: 0.00561 Median: 0.00293 SD: 0.01140

Figure 2 Predicted productivity and defect rates

Figure 2 illustrates predicted productivity and defect rates for the two selected scenarios described there. After running the model we received probability distributions and basic statistics for productivity and defect rates. In this example we can observe that we should expect in Scenario 2 a productivity rate that is 40-50% higher than in Scenario 1 and a defect rate that is 65-70% lower .

For any combinations of predictor variables the model never predicts zero defects. However, from a model where predictor variables are descriptive we should not expect predictions of defect rate of 0. In real projects the absence of defects is rather the effect of other factors, such as good process and people quality, which are not included in this part of the model. They are, however, included in the 'main' part of the model described in the first section. We treat predictions for defect rate as 'potential defect rate', which can be reduced by having good process and people.

The predicted PDRs should not be treated as the final output of the model. On the contrary, they are rather input to the 'main' part of the productivity model illustrated on Figure 1.

Summary and future work

We have developed a sub-model for estimating prior development productivity and defect rates from descriptive information. These rates are used in the main part of the productivity model for predicting quality, effort or software size. Our sub-model for estimating these rates enables us to use the main part of the model without the need to provide explicitly prior productivity and defect rates what could be problematic for some companies without the relevant data about the past projects.

Correlations between project descriptive factors and productivity and defect rates in the ISBSG dataset are consistent with the factors of productivity and defect rates in relevant literature. That proves we have a good selection of predictor variables in our model. Our sub-model provides predictions with high variance which is natural for a model predicting productivity and defect rates from descriptive data only.

We will continue the development of this sub-model and the main part of the model by adding other predictor variables using results of analyses from other sources of empirical data.

Bibliography:

1. Brooks F.P.: *The Mythical Man-Month: essays on software engineering*, 2nd edition, Addison Wesley, 1995
2. Diaz M., King J.: *How CMM Impacts Quality, Productivity, Rework, and the Bottom Line*, Crosstalk. The Journal of Defense Software Engineering, March 2002
3. ISBSG: *Estimating, Benchmarking & Research Suite Release 9*, International Software Benchmarking Standards Group, 2005
4. ISBSG: *ISBSG Comparative Estimating Tool V4.0 – User Guide*, International Software Benchmarking Standards Group, 2005

5. ISBSG: *Software Project Characteristics or Events that might impact Development Productivity*, ISBSG Special Analysis Report, 2006, http://www.isbsg.org/isbsg.nsf/weben/Downloads (accessed on 21/03/2007)
6. Liu Q., Mintram R.: *Preliminary Data Analysis Methods in Software Estimation*, Software Quality Journal, 13, 2005, pp. 91-115
7. Maxwell K.D., Forselius P.: *Benchmarking Software Development Productivity*, IEEE Software, Vol. 17, I. 1, Jan/Feb 2000, pp. 80-86
8. Maxwell K.D.: *Collecting data for comparability: benchmarking software development productivity*, IEEE Software, Vol. 18, I. 5, Sept-Oct 2001, pp. 22-25
9. MODIST: *Models of Uncertainty and Risk for Distributed Software Development*, EC Information Society Technologies Project IST-2000-28749, www.modist.org
10. Radliński Ł., Fenton N., Neil M., Marquez D.: *Improved Decision-Making for Software Managers Using Bayesian Networks*, manuscript submitted to: 11th European Software Engineering Conference (ESEC), 2007
11. Reifer D.J.: *Industry Software Cost, Quality and Productivity Benchmarks*, Software Tech News, Vol. 7, No. 2, July, 2004
12. Schacchi W.: *Understanding software productivity*, Software Engineering and Knowledge Engineering: Trends for the Next Decade, Vol. 4, World Scientific Press, 1995