PRACE NAUKOWE UNIWERSYTETU EKONOMICZNEGO WE WROCŁAWIU nr 133
Pozyskiwanie wiedzy i zarządzanie wiedzą 2010

Łukasz Radliński
University of Szczecin

INTEGRATING DATA SOURCES IN A SOFTWARE PROJECT RISK ASSESSMENT MODEL

Summary: This paper discusses integrated software project risk assessment model called the Productivity Model. There are two levels of integration involved with the model: (1) integrating effort, functionality and quality estimation and enabling trade-off analysis between these key project variables; (2) integrated knowledge base used to build the model made of various independent sources: expert knowledge, questionnaire survey results, results from statistical analysis of empirical data, and other reported data. Since the benefits of the first level of integration have been previously discussed this paper focuses on the second level by discussing the process of developing the Productivity Model using different data sources. Although this model reflects the software engineering area, such approach can be followed in building predictive models also in other fields.

Keywords: data integration, expert knowledge, risk assessment, project management, Bayesian net.

1. Introduction

Typical predictive models in various areas, including software engineering, are built using semi-or fully automated process in which developed model depends almost entirely on the dataset provided. Although such approach is feasible when the dataset of required volume and quality is accessible, it completely fails when available dataset is small, inconsistent and with missing data, as typically happens in software companies.

There are publicly available datasets (discussed in detail in [10]) which theoretically can be used to build general-purpose models. But they do not allow building models tailored for individual company needs. Thus the task to build a predictive model for a specific company when very little data is available may seem impossible. However, there is a possibility of integrating various sources of knowledge – not only using single formalized dataset. Further discussion on building general-purpose or single-company models can be found in [8].

Bayesian nets (BNs) [9] allow integrating both hard results of empirical analysis with soft expert knowledge and other sources. Such approach has been previously used to develop predictive models in various areas of software engineering such as effort estimation [2] or defect prediction [4].

This paper discusses how various sources of knowledge have been integrated in the Productivity Model – a BN for software project risk assessment. Section 2 discusses the structure of the model and its main features. Section 3 discusses how various sources of knowledge have been used to develop the Productivity Model.

2. Structure of the Productivity Model

Figure 1 illustrates the structure of the main part of the Productivity Model. Rounded rectangles reflect variables and rectangles reflect subnets – the one for process quality factors is illustrated on Figure 2. Detailed information on model structure can be found in [11; 13].

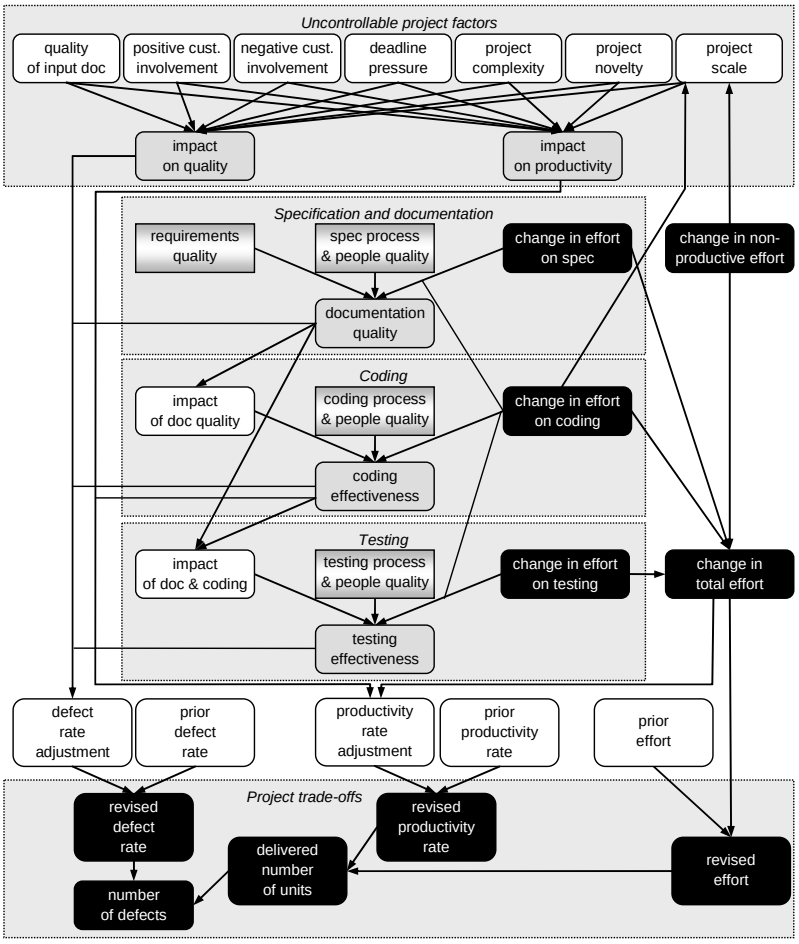


Figure 1. Structure of the Productivity Model

Source: [11].

The main goal for the Productivity Model was to enable trade-off analysis between key variables in software projects: functionality (*delivered number of units*), resources (*revised effort*) and software quality (*revised defect rate* and *number of defects*) adjusted by various process factors and effort allocation. The Productivity Model adopts basic philosophy of earlier MODIST model [2] but overcomes its main limitations and is structurally very different. The main features of the Productivity Model include:

– ability to perform trade-off analysis with variables expressed on numeric scale;
– allows entering custom prior productivity and defect rates depending on company's data from the past projects or estimate them using the PDR model illustrated on Figure 3;
– enables using different units of measurement, including custom units;
– can be easily calibrated by adjusting weights for variables using provided questionnaire;
– can be extended by adding other qualitative factors;
– numeric variables are dynamically discretised to ensure greater precision in results.

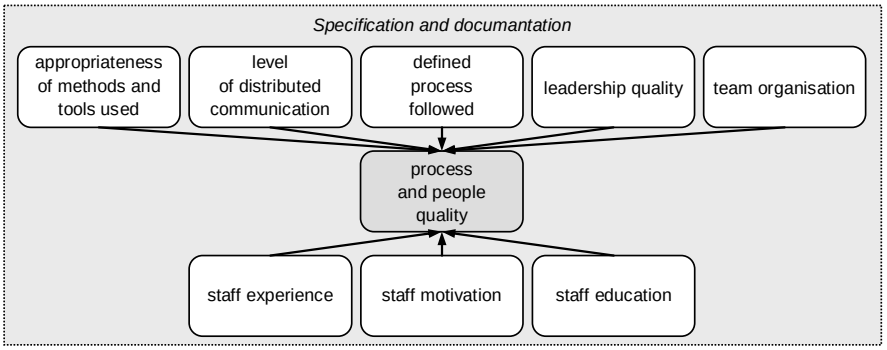


Figure 2. Structure of subnet for process and people quality

Source: [11].

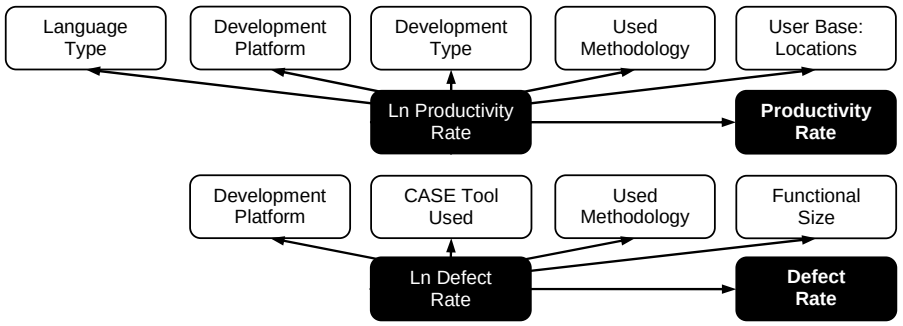


Figure 3. Structure of NBC sub-models for estimating prior productivity and defect rates

Source: [12].

3. Integration of data sources

The Productivity Model integrates various knowledge sources. Table 1 summarizes the stages of model development with knowledge sources.

Table 1. Summary of data sources at various stages of model development

Stage	Source of data
Identification of variables	literature survey, analysis of past models, expert knowledge
Model structure	expert knowledge
Priors for numeric variables	statistical analysis
Priors for ranked variables	expert knowledge
Weights for process factors	questionnaire survey
Impact of ranked variables on numeric variables	questionnaire survey

Source: own elaboration.

Model structure has been defined to reflect expert knowledge and other soft information reported in earlier studies. This stage involved building the core of the model with only minimum set of trade-off variables and successively adding new variables and subnets extending model functionality.

3.1. Priors for numeric variables

The model assumes that users from software companies can provide their own prior distribution for numeric variables (effort, functionality and defect rate). However, the model should also work when users cannot provide such data, which is actually quite common [14]. In such case the PDR model (Figure 3) estimates these distributions based on environmental factors.

These factors have been identified as predictors in the statistical analysis of ISBSG dataset [6] containing data on 3024 software projects described by 99 variables. This analysis involved the following steps [12]:

1) analyzing dependent variables and applying transformations to logged distributions;

2) preparing list of potential predictors based on experience and on analysis of factors incorporated in similar models in the past, mainly [2; 3; 4];

3) analyzing categories of potential nominal predictors which resulted in removing one predictor from further analysis (unclear interpretation) and setting new categories grouping many similar low-frequency states for some variables;

4) identifying relationships between predictors and dependent variables using Spearman's rank correlation coefficient, Kruskal-Wallis analysis of variance and box-plots;

5) identifying correlations and associations among predictors to include in the model only those predictors which are independent of each other – using the same measures as in the previous stage and additionally: Phi, Cramer's V and contingency coefficients.

The impact of predictors on productivity and defect rates reflects the results of analysis of two-way tables, box-plots and histograms. Due to missing data in the ISBSG dataset and uncertainty of the correct representation of the whole population in the dataset the impact of predictors has been adjusted by the author's expert knowledge and other data sources, mainly [7, cited after: 15].

3.2. Priors for ranked variables

Ranked variables in the Productivity Model reflect the states of uncontrollable project factors (Figure 1) and controllable process factors (Figure 2). Before explaining justification for setting prior probability distribution as was done in the current version of the model, the interpretation of these variables must be explained first. These variables are not expressed on an absolute scale as is common in other models. Instead, their state reflects the value of increase or decrease of this variable compared to the value of this variable in a project used as a template (a project developed in the past which is the most similar to the current one). The prior distribution for these variables has been set to symmetric (similar to Normal) as shown in Figure 4.

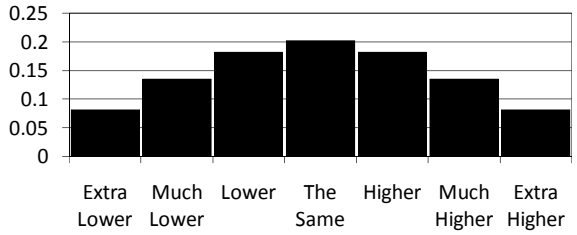


Figure 4. Prior probability distribution for ranked variables

Source: own elaboration.

Leadership quality is one of these ranked variables. When assessing the state of this variable project manager has to compare the level of *leadership quality* in the current project to the template project. Generally, the level of this variable increases over time as the leaders get more mature. But it may also drop down, for example when a manager has to put more focus to another project developed simultaneously or when a key manager has to be replaced by an inexperienced one. Moreover, the model has to take into account the fact that the template project can be either very recent one or developed long time ago. For these reasons it is sensi-

ble to assume that an increase of *leadership quality* is as equally likely as its decrease and that the level of deviation from the state "the same" is less and less probable. A similar motivation led to setting such symmetrical distribution to other ranked variables.

3.3. Weights for process factors

The final version of the model was calibrated with the results of questionnaire survey mostly experienced project managers or developers. The aim of this survey was to determine experts' opinions about the impact of relevant factors on software quality and development team productivity. Due to contradictory responses the weights have been assigned to each respondent depending on respondent's experience (as provided by each respondent) and known approach of each respondent to filling the questionnaire (motivation and time spent on thinking about providing responses).

Table 2 illustrates summarized results of this survey by presenting the mean, median and weighted mean of the values provided by respondents. The impact of predictors on dependent variables was measured from '0' (no impact) to '10' (the highest impact). The weighted mean values from Table 2 have been directly entered to the model as the weights for process factors.

Table 2. Summary of questionnaire survey results – weights for process factors

Influence on	Factor	Response value provided		
		Mean	Weighted mean	Median
1	2	3	4	5
software quality (controllable factors)	effectiveness of analysis and documentation process	8.1	7.2	8.0
	effectiveness of coding process	7.1	6.5	8.0
	effectiveness of testing process	9.1	8.3	10.0
software quality (uncontrollable factors)	project complexity	7.3	7.4	7.0
	project novelty	6.9	6.8	7.0
	project scale	5.8	5.3	7.0
	deadline pressure	8.6	8.5	9.0
	quality of input documentation	6.0	6.3	5.0
	positive customer involvement	7.3	7.2	7.0
	negative customer involvement	5.2	5.5	4.0
project group productivity (uncontrollable factors)	project complexity	7.1	7.1	8.0
	project novelty	7.9	8.0	7.0
	project scale	5.7	5.9	7.0
	deadline pressure	7.3	7.0	8.0
	quality of input documentation	5.6	6.1	5.0
	positive customer involvement	6.2	5.4	7.0
	negative customer involvement	5.2	4.8	4.0

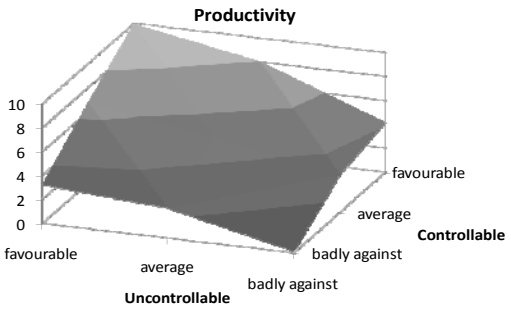
1	2	3	4	5
overall process and people quality	staff experience	8.4	8.3	9.0
	staff motivation	8.8	8.7	9.0
	staff education	6.6	6.7	6.0
	team organisation	7.8	7.8	8.0
	appropriateness of methods and tools used	7.8	7.4	8.0
	level of distributed communications	6.9	6.8	8.0
	well defined process followed	6.2	5.8	6.0
	leadership quality	7.3	7.2	7.0
effectiveness of analysis and documentation process	requirements quality	8.3	8.5	9.0
	process and people quality	6.9	6.9	8.0
	effort	7.7	7.8	8.0
effectiveness of coding process	documentation quality	7.0	6.6	8.0
	process and people quality	7.7	7.9	8.0
	effort	6.9	6.8	7.0
effectiveness of testing process	documentation quality	7.9	8.1	8.0
	process and people quality	7.6	8.1	8.0
	effort	7.6	8.3	8.0

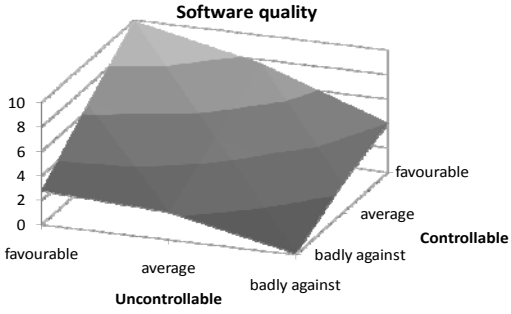
Source: [11].

3.4. Impact of ranked variables on numeric variables

Figure 5 illustrates respondents' opinions on the expected change in productivity and software quality depending on different combinations of controllable and uncontrollable factors. The scale for these changes is from '0' to '10'. Value '0' indicates the greatest possible decrease in productivity and software quality, value '5' means no change and value '10' indicates the highest possible increase.

To incorporate results from Figure 5 it was necessary to find the expressions which best suit the gathered data. When analyzing the perceived impact of controllable and uncontrollable factors on productivity and software quality it turned out that these relationships are most accurately reflected by the *weighted min* function [5] – a combination of min and weighted mean functions. The weights in this function have been estimated using Generalized Reduced Gradient algorithm implemented in Microsoft Excel's Solver tool by minimizing the mean magnitude of relative error (MMRE) between the actual values provided by respondents and the estimates produced by the *weighted min* function. Table 3 summarizes these estimated weights.





		Uncontrollable		
		favourable	average	badly against
Controllable	favourable	10	8.0	4.1
	average	6.7	5	3.1
	badly against	3.2	2.5	0

		Uncontrollable		
		favourable	average	badly against
Controllable	favourable	10	7.6	4.0
	average	6.4	5	2.4
	badly against	2.7	2.2	0

Figure 5. Impact of controllable and uncontrollable factors on productivity and software quality

Source: [11].

Table 3. Weights for controllable and uncontrollable factors

Dependents / Predictors	Productivity	Software quality
Controllable factors	2.422	3.615
Uncontrollable factors	1.581	1.667

Source: [11].

Table 4 summarizes how much productivity and software quality are expected to decrease or increase should a combination of all possible factors be least or most favourable according to the respondents. For example, when all factors are least favourable, project group productivity is expected to decrease to 0.21 of its typical value; when all factors are most favourable, project group productivity is expected to increase to 5.63 times higher of its typical value.

Equations (1)-(4) define overall combined impact of both controllable and uncontrollable factors on effort and productivity and defect rates by taking into account results fromTable 4.

Table 4. Summary of questionnaire survey results – impact of ranked variables on numeric variables

Influence on	Factor	Response value provided		
		Mean	Weighted mean	Median
software quality	all factors overall – worst possible	0.19	0.14	0.10
	all factors overall – best possible	5.94	5.95	5.00
project group productivity	all factors overall – worst possible	0.23	0.21	0.21
	all factors overall – best possible	5.56	5.63	4.00

Source: [11].

$$revised\_effort = prior\_effort \times 10^{4 \times change\_in\_total\_effort - 2}, \quad (1)$$

$$prod\_dummy = prior\_productivity\_rate \times 2.4^{4 \times productivity\_rate\_adjustment - 2}, \quad (2)$$

$$revised\_productivity\_rate = prod\_dummy \times 10^{-4 \times change\_in\_total\_effort + 2}, \quad (3)$$

$$revised\_defect\_rate = prior\_defect\_rate \times 2.6^{-4 \times defect\_rate\_adjustment + 2}. \quad (4)$$

With these expressions the model assumes that *revised defect rate* can change from 0.15 to 6.76 of the *prior defect rate* and that *revised productivity rate* ranges from 0.17 to 5.76 of the *prior productivity rate*. The model calculates the *revised productivity rate* in two steps because it depends on *change in total effort* and *change in other factors*.

4. Conclusion and future work

Integrating various sources of knowledge in developing predictive models for software engineering appears to be a useful approach when companies do not have enough high-quality empirical data in a single dataset to use automated learning methods. Integrating expert knowledge with results of empirical analyses enables developing models tailored for individual company needs based on solid empirical results. However, still extreme caution should be taken when using the model outside the original scope of the data [1].

Future work in this area will involve extending the Productivity Model by developing a sub-model for estimating prior effort based on environmental factors and by incorporating characteristics of reused code from previous projects. A significant useful extension of the Productivity Model would be to transform it from a BN to an influence diagram and therefore enable performing optimization of the project management plan. However, to do this properly either new algorithms for calculating influence diagrams with continuous decision nodes should be developed or merging an influence diagram with other optimization techniques such as genetic algorithms or multi-criteria programming.

Acknowledgements

I would like to thank Prof. Norman Fenton (University of London) for his helpful guidance in developing the Productivity Model, Dr. David Marquez (University of London) for his support in performing statistical analysis, Thomas Schulz (Robert Bosch GmbH) for help in questionnaire preparation and all anonymous individuals who acted as respondents in the questionnaire survey.

References

[1] Druzdzel M.J., Diez F.J., *Combining knowledge from different sources in causal probabilistic models*, "Journal of Machine Learning Research" 2003, Vol. 4, pp. 293-316.

[2] Fenton N., Marsh W., Neil M., Cates P., Forey S., Tailor M., *Making resource decisions for software projects*, [in:] *Proceedings of 26th International Conference on Software Engineering* (May 23-28, 2004), IEEE Computer Society, Washington, DC, 2004, pp. 397-406.

[3] Fenton N., Neil M., Marsh W., Hearty P., Radliński Ł., Krause P., *Project data incorporating qualitative factors for improved software defect prediction*, [in:] *Proceedings of 3rd International Workshop on Predictor Models in Software Engineering. International Conference on Software Engineering* (*May 20–26, 2007*), IEEE Computer Society, Washington, DC, 2.

[4] Fenton N., Neil M., Marsh W., Hearty P., Radliński Ł., Krause P., *On the effectiveness of early life cycle defect prediction with Bayesian Nets*, "Empirical Software Engineering" 2008, Vol. 13, No. 5, pp. 499-537.

[5] Fenton N.E., Neil M., Caballero J.G., *Using ranked nodes to model qualitative judgements in Bayesian networks*, "IEEE Transactions on Knowledge and Data Engineering" 2007, Vol. 19, No. 10, pp. 1420-1432.

[6] ISBSG, *Estimating, Benchmarking & Research Suite Release 9*, International Software Benchmarking Standards Group, 2005, www.isbsg.org.

[7] Jones C., *Software Quality in 2002: A Survey of the State of the Art*, Software Productivity Research, Inc., 2002.

[8] Mendes E., Lokan C., *Replicating studies on cross- vs single-company effort models using the ISBSG Database*, "Empirical Software Engineering" 2008, Vol. 13, No. 1, pp. 3-37.

[9] Pearl J., *Probabilistic Reasoning in Intelligent Systems*: *Networks of Plausible Inference*, Morgan Kaufmann, San Francisco 1988.

[10] Radliński Ł., *Przegląd publicznie dostępnych baz danych przedsięwzięć informatycznych*, "Studia Informatica', Vol. 22 [forthcoming].

[11] Radliński L., *Improved Software Project Risk Assessment using Bayesian Nets*, Ph.D. Thesis, Queen Mary, University of London, London 2008.

[12] Radliński Ł., Fenton N., Marquez D., *Estimating productivity and defects rates based on environmental factors*, [in:] *Information Systems Architecture and Technology: Models of the Organisation's Risk Management*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2008, pp. 103-113.

[13] Radliński Ł., Fenton N., Neil M., Marquez D., *Improved decision-making for software managers using Bayesian Networks*, [in:] *Proceedings of 11th IASTED International Conference on Software Engineering and Applications*, Cambridge, Mass., 2007, pp. 13-19.

[14] Rainer A., Hall T., *Identifying the causes of poor progress in software projects*, [in:] *Proceedings of 10th International Symposium on Software Metrics*, September 2004, pp. 184-195.

[15] Sassenburg J.A., *Design of a Methodology to Support Software Release Decisions* (*Do the numbers Really Matter*?), PhD Thesis, University of Groningen, Groningen 2006.

INTEGRACJA ŹRÓDEŁ DANYCH W MODELU DO SZACOWANIA RYZYKA PRZEDSIĘWZIĘCIA INFORMATYCZNEGO

Streszczenie: Niniejszy artykuł poświęcony jest zintegrowanemu modelowi do szacowania ryzyka przedsięwzięcia informatycznego, nazwanego modelem produktywności. W modelu występują dwa poziomy integracji: (1) zintegrowanie szacowania nakładów, funkcjonalności i jakości umożliwiające analizę kompromisu między tymi kluczowymi zmiennymi projektu; (2) zintegrowana baza wiedzy użyta do budowy modelu złożona z różnych niezależnych wzajemnie źródeł: wiedzy eksperckiej, wyników badania ankietowego, wyników analizy statystycznej na danych empirycznych i innych opublikowanych danych. Zalety pierwszego poziomu integracji zostały przeanalizowane wcześniej, dlatego niniejszy artykuł skupiony jest wokół drugiego poziomu – procesu budowy modelu produktywności z różnych źródeł danych. Model ten odzwierciedla obszar inżynierii oprogramowania, jednak proponowane podejście może być wykorzystane do budowy modeli prognostycznych również w innych dziedzinach.