Economic model to determine the lowest price limit of fixed price contracts in software engineering

David Kuhlen[1], Andreas Speck[2], and Dennis Pfisterer[3]

[1]IU International University of Applied Sciences, Waterloohain 9, 22769 Hamburg, Germany
[2]Christian-Albrechts-Universität zu Kiel, Head of the Business Information Technology Group, Hermann-Rodewald-Straße 3, 24098 Kiel, Germany,

[3]Duale Hochschule Baden-Württemberg, Coblitzallee 1-9, 68163 Mannheim, Germany

Abstract. Competition forces software producers to offer their goods and services at the lowest costs in practice (49, p. 3), (4, p. 2). In this case, software errors can lead to additional processing costs which has to be covered by the software producers themselves (20). To compensate this, software manufacturers often have the option to cross-subsidize low-priced offer prices with offers billed hourly, e.g. on the basis of service contracts. However, the relation of possibilities for cross-subsidization with the risk of software errors is not clearly predictable by the process model (52, p. 9). As this relation is defined by the process, the aim to reach cost-control and cost-transparency is strongly related to the aim of (process) improvement (51, p. 39). In order to facilitate software producers within the calculation of minimum prices, required to cover the costs, an economical prediction model will be presented in this paper. This model bases on a simulation experiment, consisting of multiple scenarios. The scenarios were derived by a variation of the risik probability and the possibility of cross-subsidization.

1 Motivation and Contribution

The current state of research confirms that the production of software has the potential to gain the highest profitability (44, p. 145). However, in practice the competition forces software producers to offer their services at the lowest costs sometimes, because customers seek for a cost reduction (39; 18).

An increasing proportion of IT-costs of the revenues (40, p. 2) leads customers to care for such a cost reduction. In order to cover their costs, software producers could enter into different types of contracts, regarding to the terms of payment (e.g. fixed or hourly). Normally, a *contract to produce a work* or a *contract for surrender of use* will be made in the case of buying a software (15, pp. 156 – 157).

The type of the contract often defines the terms of payments. Normally, the parties agree on billing at an hourly basis in service contracts, whereas in *contracts to produce a work* a fixed price will be appointed. During the last decades customers asked for fixed-price-contracts at increasing intervals, in order to reduce the uncertainties of software production (26, p. 17). This leads to financial risks if additional expenditure of times (e.g. because of software faults) occur which is not accountable. Software producers are forced to calculate a minimum price which is able to cover such costs, depending on the risk.

This work investigates the calculation of a critical selling price for software products. The production of software is characterized by fixed costs (41; 7). As a result, variable costs are nearly non-existent (46, p. 412). Therefore, short time calculations of the lowest price which is limited to the variable costs are, de facto, meaningless. Finally, the different kinds of gross margins won't lead to a useful price calculation. The gross margin I

DOI: 10.5121/ijsea.2024.15402

will be zero, whereas gross margin II or III are defined by the full personal costs, under normal production circumstances. Therefore, the calculation of the lowest price on a short time basis shows that the costs have to be anywhere between zero and the full personal costs. This doesn't give any gain of knowledge. In order to support the management, a new prediction is needed. Facilitating the management which has the most influence on operational dimension (28) seems to contribute the financial success directly.

Furthermore, such calculation of the lowest price doesn't care for the configuration of the production capacity, possibilities of cross-subsidization or cyclic risks, because of software faults. In order to facilitate the IT-Management, an economic model which considers these factors will be elaborated in this work. This model bases on the calculation of the maximum productivity comparing to the investigations in the field of the cost-reduction potential function (34). On the basis of the maximum productivity, the hypothetical minimum price could be calculated. In order to test the impact of cross-subsidization the price calculation has fallen short of this minimum price.

This way, this work contributes to answer the following question: *How much could the offer price fall short of the minimum price, considering possibilities of cross-subsidization and cyclic risks?*

The contribution of this work is given by its mathematical concept which increases the cost transparency, necessary, in order to calculate the prices of IT services (45, p. 60). In the following, a calculation procedure is shown in an exemplary way which bases on assumptions. A huge number of interactions within the ecosystem of software engineering have an influence on the price limit in practice (43). It should be pointed out that the numeric values of the following results are just exemplary. Because of the fact that no universal recommendation could be made, suitable for any economic situation, the presented concept has to be transferred and adapted before its use in practice.

2 Related Work

Even if the question in section 1 has not been answered already, there are other studies answering related questions. For example, the field of cost factors has been investigated intensively. One of the earliest publications on cast factors was published by Farr und Zagorski in 1964 (21). Their work describes the data collection and its analysis in order to elaborate a formula system which determines the relation of cost influence factors (21).

The relation of cost influence factors is often explained by classical economic models. Therefore, cost models which allow internal cost allocations, such as described by Götze (23, p. 1), are the basis to explain the relation of cost influence factors. Furthermore, the explanations of Gutenberg regarding the cost- and production function found a basis to elaborate economic prediction models (25, p. 366). The results of Gutenberg were used by Knolmayer in order to investigate the validity of Brooks findings (14), regarding to the impact of capacity-configurations on the performance of the development procedure (33).

This leads to mathematical models containing factors which influence the progression of costs. One of such models is called the *software function* which was formulated by Albrecht and Gaffney in 1983 (3). Further models were developed by Boehm (12) or Abran et al. (1, p. 104). The basis of such parametric estimation models is to determine the required effort depending on parameters which explain the complexity of the task or situation within the environment. As to that, Bailey and Basili mention the limitation of environmental factors to be one of the key problems (6, p. 115).

Both to determine the statistical significance and for the identification of formulas which interrelate the factors, a regression analysis turns out to be sensible. Therefore, Walston and Felix came up with a regression analysis in order to determine the productivity of software engineering (53). Furthermore, Do and Rothermel conducted a regression analysis to simplify a model of cost influence factors (19). Next to the construction of cost models, a regression analysis is used to estimate the software production effort. For this purpose, regression models were developed by Jørgensen (29, p. 297) or Adalier et al. (2) for example.

Investigations in the field of the software price have received less attention in the last decades, as Prud'homme et al. explained (42, p. 1001). During their investigations on the progression of software prices, Prud'homme et al. observed declining prices for business software (about 4.4 %) and for consumer software (about 7.9 %) (42, p. 1001). Furthermore, the field of software price was mentioned in the investigations of Lucas et al., which explains how high prices indicate low skills of the customers (38). One of the most famous publications on software price calculation is given by Kittlaus and Clough. They explain different ways in order to determine the price of a software product, regarding to aspects like costs or intellectual property (32).

Furthermore, this work builds upon investigations on the contractual form, used in software engineering. In this context, the results of Buhl on contractual forms which are used in the procurement are relevant to mention (15, p. 161). Regarding to the software engineering, Wolf explains the impact of agile development approaches on the contractual form (54, p. 25). In addition, also Becker explains challenges which show that the contract needs to match the process, used in the development approach (8). Next to Becker, Gennen revealed that the contractual form has to be aligned to the procedural model in order to define warranties economically (22, p. 41). In order to reduce such risks, Benaroch shows the possibility of subcontractings (9).

Regarding to the procedural development approach, Boehm recommended to relate the progression of software development with payments (13, p. 70). Even if this is not equal to payments on an hourly basis, the recommendation offers some advantage regarding to the solvency of software producers. It connects the terms of payments to the states of the development procedure. This offers the potential of interesting calculations, considering the explanations on the formal nucleus of business process models given by Speck (47, p. 427). In further investigations this might be helpful, to consider the time preference of payments within the model.

3 Method

In order to answer the question, given in section 1, we have to elaborate a model that captures the situation of order management for software producers. In a short form (comparing to (34)), the development process might consist of the activities *Requirements Engineering*, *Development* and *Delivery*. In these activities customer requirements get analysed and a software solution is getting developed and delivered to the customer. The *quality assurance* is part of the *development* in this simple process models. Furthermore, as presented in (34), each activity is processed by one group of actors (*Product Owner, Software engineers* and *Support*). However, also in this model the *group-distinction-property* (see Rule R15, (34)) has to be fulfilled.

As the development of software requires a close cooperation with customers, the customers have to be part of the model, too. Therefore, another group of actors contains all customers. In practice this cooperation causes a social complexity (35) for example if software producers have to decide between different requirements (48, p. 1). As explained in (34), the model of this investigation is restricted to aspects of capacity configuration of the development process. Social-interactive aspects are ignored, therefore.

As described in (37), the customers create requirements. In order to fulfil these requirements, a contract form is chosen. Regarding to this contractual form, the terms of payment define whether customers pay a fixed price. Otherwise, customers may agree on paying for the services on an hourly basis. After the development is finished, the customer checks the product in his acceptance test. Faults lead to a cycle, because software producers have to execute the activity *development* once more, in order to fix the bug.

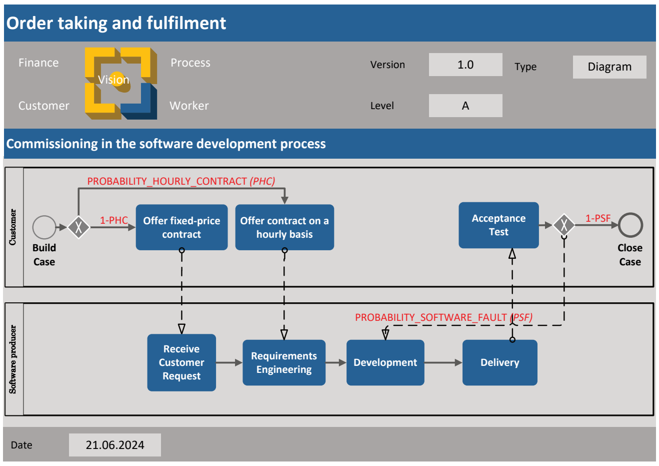

Fig. 1. Model of the software production process to investigate the impact of different commisioning strategies, own illustration created with (17)

In 1 the model of the software development process is illustrated with its activities, concerning the agreements on the contractual form. On the basis of this model which is an abstraction (30), the leading question of section 1 could be answered. Therefore, the simulation software *PAS*, presented in (34; 37; 36), is used to perform an experiment. During this experiment, multiple simulations were performed in order to analyse different scenarios. These scenarios differ, regarding to the parameters of the process, illustrated in 1. For each scenario the result gets measured after its simulation. To measure the result of a simulation, the calculation of the accounts receivables of a software producer is meaningful.

The accounts receivables result from the product of worked hours accountable with its hourly rate. Different groups of actors might have different charge out rates. Furthermore, the amount of accounts receivables contain the earnings, expected from fixed price contracts. Optimistic it might be assumed, that the fixed price is accountable immediately when the software producer takes the order. Concerning this matter, it has to be considered that software producers have to fulfil the contract, until they have a right to claim the fixed price in practice. In service contracts, where both parties agree on the payment on an hourly basis, this right is acquired after the software producer spends his work hour.

In order to investigate these scenarios by the use of the software *PAS*, five enhancements are necessary. First, new *KPIs* have to be introduced in order to define the fixed prices and the hourly rates of actor groups. Second, a *KPI* has to be introduced which contains the amount of open accounts receivable that belongs to a case. This KPI makes sure that the fixed price is accounted just for one time. Furthermore, a KPI is necessary which contains the amount of accounts receivables of an organisation (OAR). Finally, a procedure is needed which calculates the amount of accounts receivable (OAR) on the basis of the worked hours spent and the fixed prices of orders, taken by the organisation.

The procedure accounts receivables depending on the amount of worked hours. This amount depends on the probability of agreements for paying on an hourly basis (PHC). Furthermore, the probability of software faults (PSF) has a double influence. The lack of a clear specification on the software to be delivered which is typical if customers and software producers agree on an hourly payment, leads to more accountable worked hours. Ignoring the risk of damages to the image of software producers, this increase of worked hours is valuable. If software producers and customers agree on a fixed price instead, worked hours are not accountable. Therefore, a high PSF leads to financial losses in such contracts.

To simplify the terminology, agreements between software producers and customers which require the payment of a fixed price, are named to be *contracts for work (CfW)*. On the opposite, agreements on payments on an hourly basis are named to be *service contracts (CfS)*. It has to be mentioned that different contractual forms do not have to differ in the terms of payments automatically. However, CfW normally requires for the construction of a product, whereas CfS focuses on the work hours performed. Therefore, the differentiation of CfS and CfW seems to be acceptable in order to simplify the spelling.

In order to answer the leading question, scenarios will be analysed which consist of a variation of the parameters, showed in List L1. This method seeks to minimize the risk of simulating unnecessary KPIs described by Radonjic-Simic, Wolff and Pfisterer (43).

- PHC: Variation of the probability of service contracts in the interval of 0% to 70 %
- PSF: Variation of the risk of software faults in the interval of 0% to 50 %
- VMPr: Variation of the shortfall of the minimum fixed price (MFP) in the interval of 50 % to 100 %

List L1: Variation of parameters to build the scenarios

The list given in List L1 shows the necessity of a minimum fixed base-price to be calculated which is deceeded on a special percentage (VMPr). The resulting minimum fixed price (MFP) is used in the simulation and will be assigned as *vertex accountable price* in the model. The minimum fixed base-price is a result of the maximum productivity. In order to determine the maximum productivity, some assumptions on the variable and constant cycle times, the personal capacities and the monthly payroll costs are necessary, as shown in Table T1. Multiplying the personal capacity with its costs leads to the total personnel costs. Furthermore, the process effort could be calculated, considering the risk of cyclic repetitions (34). In order to calculate the maximum productivity on a theoretical basis, no cyclic repetitions will be assumed (optimistic PSF = 0%). Therefore, the rules R38 and R40, described in (34), do not have to be used in order to calculate this optimistic assumption on the base-price.

However, the delay, resulting from the procedural order of activities, has to be considered. As the development needs the requirements engineering to be finished before it starts, the production capacity of the development and the support have to be reduced by the productivity (outcome) of the requirements engineering. Furthermore, the division of the adjusted production capacity available by the effort required to produce 1 FP results in the maximum number of producible cases per team and month. The bottleneck limits the total productivity to the minimum, with a small exception.

In theory, the process configuration shows in List L1 that software developers could produce 87 FP per moth, the product management could produce 90 FP per month and the support is able to produce 122.31 FP per month. As the outcome of the support is limited by the outcome of the software development (compare the order of the sequence flow in 1), the support is limited to just being able to produce 87 FP per month. However, the product management is able to produce cases which are unable to be completed during the month, because of limitations by software developments capacity. Also if cases could not be finished during the month, they may lead to a payment claim, because of the above mentioned assumptions. Therefore, the capacity of the product management is not limited by the bottleneck within the model.

In order to produce the maximum outcome, the product management requires 435.48 h, the software development requires 348 h and the support requires 94.25 h. To cover the personal costs, a work hour needs to cost 22.04 € in the product management, 30.17 € in the software development and 26.53 in the support. However, there are more fixed costs which have to be covered, like e. g. insurance costs, marketing costs or taxes. These costs are highly individual by different software producers. In order to cover these costs and to earn a net income, an addition of 70 % is calculated (*assumption*). By doing so, the price per work hour raises up to 73.48 € in the product management, 100.57 € in the software development and 88.42 € in the support.

	Requirements Engineering	Software-Development	Delivery	
variable effort	290	200	60	min
constant effort	30	120	15	min
number of employees	3	3	1	pc
personnel salary per month	3200	3500	2500	€
total personnel costs	9600	10500	2500	€
minutes to produce 1 FP	320	320	75	min
hours to produce 1 FP	5,33	5,33	1,25	h
availabe production capacity	28800	28800	9600	min
adjusted production capacity	28800	27840	9173,33	min
Max. FP (producible)	90	87	122,31	FPs
Max. FP (adjusted)	90	87	87	FPs
hours of production	435,48	348	94,25	h
costs per production hour	22,05	30,17	26,53	€ / h
price per production hour	73,48	100,57	88,42	€ / h

Table 1: *Expected effort and productivity .*

However, the minimum fixed base-price is calculated without the addition, required to cover fixed costs (compare for explanations regarding to gross margin I and II, e. g. by (31)). In order to calculate the base-price, the expected effort of 1 FP (considering the maximum productivity) is multiplied with the costs per work hour and the results are summed up. This leads to a minimum fixed base-price of 311.65 € per FP, necessary just to cover the personnel costs. In each scenario this base-price will be deceeded, whereas the shortfall varies within the interval of 50 % to 100 %.

4 Data

During the utilization of the procedure presented in section 3, the simulation experiment was executed 356 times ($n = 356$). Each simulation can be described by a set of specific

KPIs. First of all, simulations differ regarding to the scenario variables "probability for offers on an hourly basis" *(PHC)*, "probability of software faults" *(PSF)* and the "reduction of the minimum fixed price" *(VMPr)*, described in section 3. Furthermore, the MFP (used minimum fixed price) is calculable which belongs to each VMPr.

The key performance indicators PSF, PHC, MFP and VMPr parametrize the simulation model before its execution. Other key performance indicators like the personal capacity or its costs were not varied. It has to be mentioned that a variation of team size is also able to contribute to a cost reduction (10, p. 1784), (11), (49), (34) which facilitates the pricing, too.

After its execution, the result of a simulation can be described by four KPIs. The amount of accounts receivables express the success of a software producer, as described in 3. Therefore the OAR (organisation accounts receivables) is calculated cyclic. On the basis of this KPI, the net income can be calculated. In order to analyse this value, the *profit contribution a (PCa)* is defined to be $PCa := OAR - TPC$.

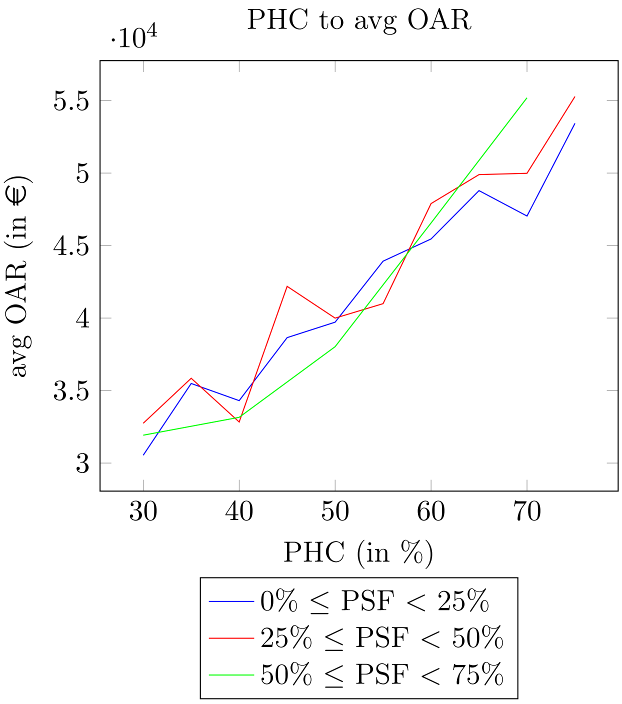

Fig. 2. Impact of PHC on avg OAR, own illustration created with TikZ (tikzpicture) being part of TeX Live Version 2024 (50) on the basis of data sets extracted from the simulation software and proprocessed with (16)

In order to determine the expected balance after cost allowance, *PCb* has to be calculated. This calculation bases on assumptions on the expected fixed costs. Therefore, the income statements of multiple software producers in Germany, available at Bundesanzeiger, have been analysed. In each income statement the amount on fixed costs (without personal costs) was calculated and related to the turnover. Investigations show an average quote of fixed costs to the turnover to be about 33.47 %. This enables the calculation of PCb as following $PCb := (OAR * 0.6653) - TPC$. The expected net income PCb enables to calculate the profit margin (PM) which is defined to be the return of turnover $PM := PCb/OAR$.

By measuring the variables mentioned above, the influence of PHC on the average OAR could be measured. Figure 2 shows the results of this analysis. Different levels of PSF are distinguished by different curves within the graphic. Graphic 2 shows that increasing an increasing number of cases which are payed on an hourly basis lead to an increase of OAR. It appears to be remarkable that the curves of different probabilities of software faults are close together. This suggests the software faults impact to be less than the impact of PHC.

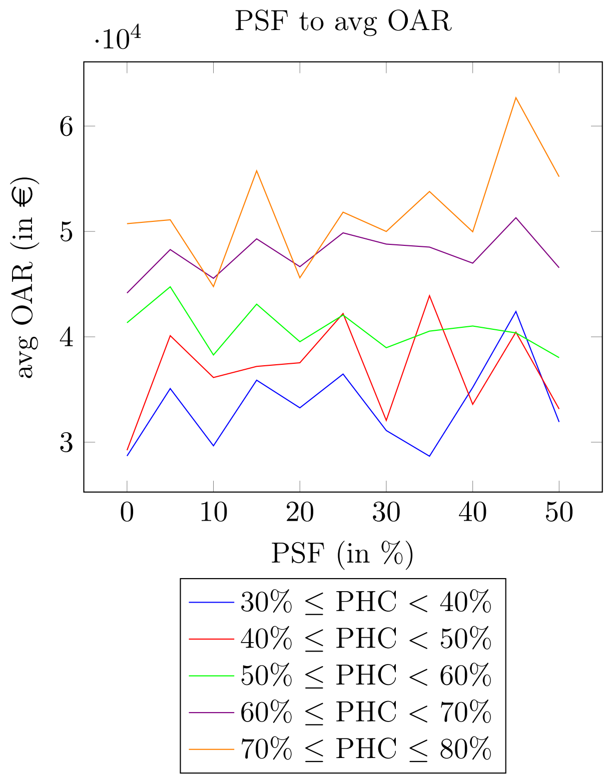

Fig. 3. Impact of PSF on avg OAR, own illustration created with TikZ (tikzpicture) being part of TeX Live Version 2024 (50) on the basis of data sets extracted from the simulation software and proprocessed with (16)

Figure 3 shows the impact of software faults on the amount of accounts receivables in detail. The graphic illustrates the progression, the amount of accounts receivables takes, influenced by PSF. Different curves show the impact, regarding to the probability of service contracts (PHC). First, figure 3 seems to confirm the assumption on PSF having a lower impact than PHC. If the probability of CfS is high (=high PHC), PSF seems to have a lower impact on OAR. However, if the probability of PHC is low, OAS is fluctuating more. A high risk of software faults (PSF) does not automatically harms the amount of accounts receivables. In order to explain the correlation, a mathematical model is needed.

Figure 4 shows the impact of VMPr on OAR. Within the graphic, different curves show the impact regarding to the probability PHC. If the probability of service contracts is high, VMPr has a minor impact on OAR. This shows the possibility of software producers of cross-subsidization. However, if service contracts play a minor part, an increasing VMPr leads to a reduction of OAR. Therefore, PHC is limited to be able to compensate the impact of VMPr.

Figure 5 shows the impact of VMPr on OAR, regarding to PSF. The graphic shows an increasing VMPr to cause a decrease of OAR. Like figure 3, also figure 5 shows an increase of the risk probability PSF which leads to a greater fluctuation of OAR if PHC is low.

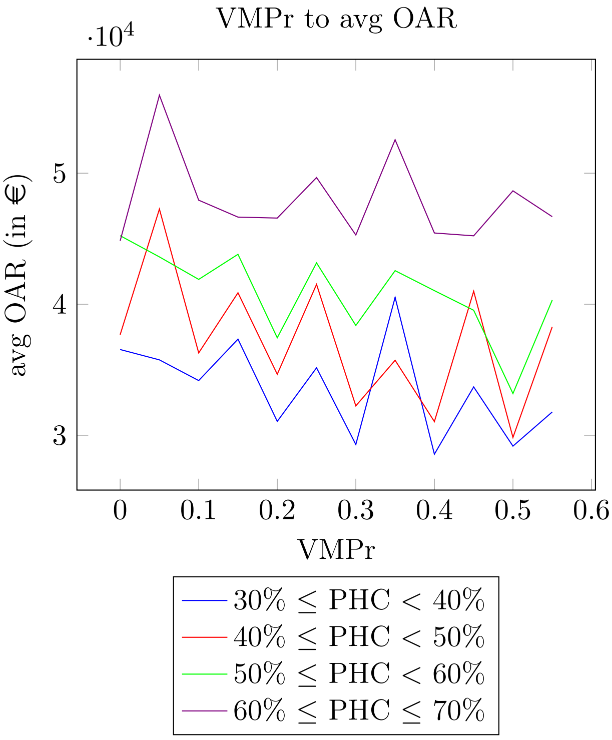

Fig. 4. Impact of VMPr, influenced by PHC on avg OAR, own illustration created with TikZ (tikzpicture) being part of TeX Live Version 2024 (50) on the basis of data sets extracted from the simulation software and proprocessed with (16)

Once more this circumstances lead to the expectation of software faults having a minor impact on OAR.

Next to investigations on the impact of VMPr, PSF or PHC on OAR, the progression of the net income has to be analysed. For this reason, figure 6 relates VMPr to the average PCb. To maximize the transparency, curves are differentiated regarding to PHC in figure 6. The figure shows that a low rate of service contracts causes an increasing VMPr. However, if PHC is high, an increased price reduction has nearly no impact on PCb. Furthermore if the rate of service contracts is less than 50 %, a shortfall on the minimum fixed base-price might cause financial losses.

In order to get comparable results, an abstraction of the results from the concrete amount of the surplus is required. For this purpose, the calculation of the return on sales was established. In line with this, VMPr is set in relation to the average return on sales PM. Within the figure, curves are differentiated regarding to PHC. The illustration in figure 7 shows that a PHC within the interval $[30\%; 40\%[$ leads to a negative return on sales if VMPr is higher than 5 %. However if the amount of service contracts is high (PHC in the interval $[60\%; 80\%]$), a variation of VMPr has nearly no impact on PM.

5 Results

The investigations in section 4 suggests a multiple correlations within the data. In order to answer the leading question, described in section 1, a regression analysis is performed. This analysis aims to determine the impact of PSF, PHC and VMPr on the depending variables OAR, PCb and PM. Within the regression analysis, a linear approximation is calculated

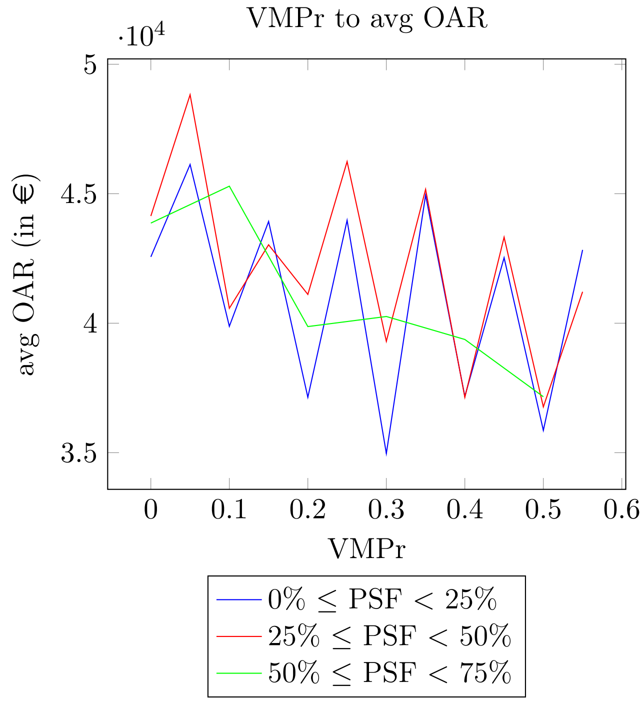

Fig. 5. Impact of VMPr, influenced by PSF on avg OAR, own illustration created with TikZ (tikzpicture) being part of TeX Live Version 2024 (50) on the basis of data sets extracted from the simulation software and proprocessed with (16)

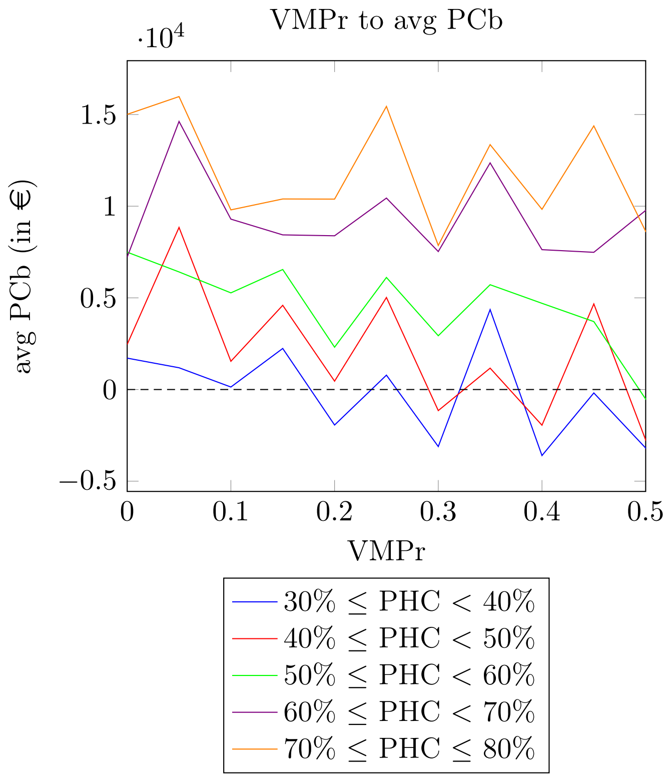

Fig. 6. Impact of VMPr, influenced by PHC on avg OAR, own illustration created with TikZ (tikzpicture) being part of TeX Live Version 2024 (50) on the basis of data sets extracted from the simulation software and proprocessed with (16)

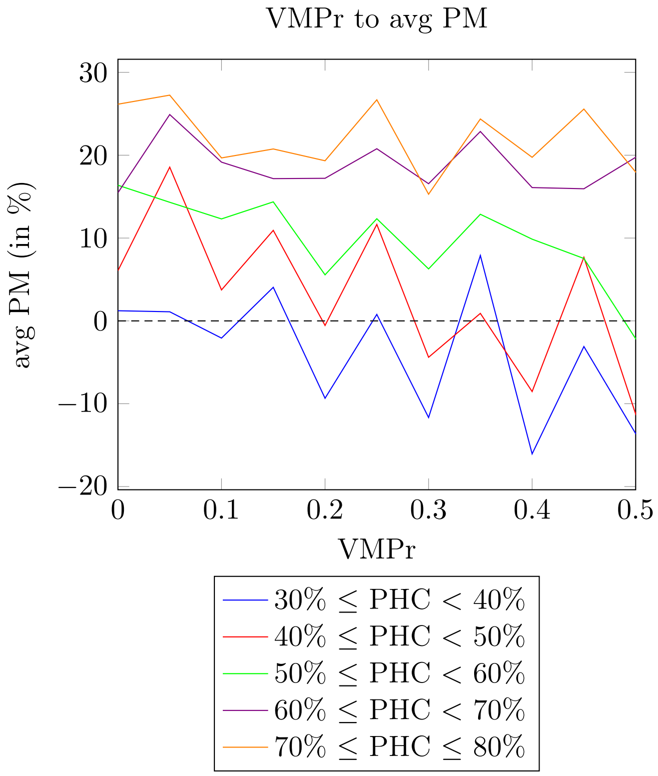

Fig. 7. Impact of VMPr on avg PM, influenced by PHC, own illustration created with TikZ (tikzpicture) being part of TeX Live Version 2024 (50) on the basis of data sets extracted from the simulation software and proprocessed with (16)

that subtends the origin (so zero is a constant). In order to determine the significance of the regression, the adjusted coefficient of determination R^2_{adj} is calculated (see Rule R1).

It has to be pointed out that a non-linear regression may lead to results, being more accurate. On the basis of the underlying economic model, it seems to be expectable that an exponential function could result in data which explains the correlation more precisely. However, Armbrust et al., Green and Rosemann or Hammer and Champy (5; 24; 27) suggest keeping a model simple in general if it should be applicable. Therefore, a linear regression is focused in order to derive the first version of this prediction model, because its application is easier. In later versions, other forms of regressions should be tested.

Rule R1: *Calculation of the adjusted coefficient of determination*

In order to calculate the adjusted coefficient of determination (R^2_{adj}), a regression of the independent variable x to the dependent variable y has to be analysed. Let n be the number of observations and $s_{x,y}$ be the standard deviation of x, y, defined as $\sqrt{\sum_{i=1}^{n} \frac{(x_i - \overline{x})^2}{n}}$, whereas $\overline{x, y}$ are the average values of the variables. On the basis of the covariance $Cov(x, y)$ which is defined as $\sum_{i=1}^{n} \frac{(x_i - \overline{x})(y_i - \overline{y})}{n}$, the adjusted coefficient could be calculated as follows:

$$R^2_{adj} := 1 - \left(1 - \left(\frac{Cov(x, y)}{s_x \cdot s_y}\right)^2\right) \cdot \frac{n-1}{n-p-1}$$

The adjusted coefficient of determination is in the interval of $[0; 1]$. The closer R^2_{adj} comes to 1, the stronger is the correlation and therefore the validity of the regression.

5.1 Regression 1: PSF, PHC and VMPr → OAR

First, the impact of PSF, PHC and VMPr on OAR are investigated. The results of the regression analysis are shown in table T2. The calculation of the regression leads to a multiple correlation coefficient of being 0.984 and a coefficient of determination of being 0.968. On this basis R^2_{adj} is 0.965. By performing an ANOVA-analysis, the regression yields a F-statistic to be 3617.614 on the basis of 3 degrees of freedom. This amount proves the statistical significance of the regression.

Table 2: Result of the regression analysis, calculated with (16)

variable	coefficient	standard error	t-statistics	p-value
PHC	730,20	15,98	45,72	0,00
PSF	153,29	24,07	6,37	0,00
VMPr	-1597,11	2169,04	-0,74	0,46

Table 2: *Results of regression 1 .*

The results in table T2 show the impact of independent variables on the quality of regression 1. As indicated by the values of the *t-statistics*, PHC and PSF contribute to the quality of the model significantly. However, the calculations revealed a *t-statistics* of VMPr to be -0.736. Therefore, the null hypothesis (h0), where VMPr has no influence on the regression, is not declinable. Furthermore, the standard error regarding to the variable VMPr is remarkably high. Consequently, VMPr has to be removed from regression 1.

All in all, regression 1 yields to be a valid model which enables software producers to estimate the amount of OAR on the basis of PHC and PSF. The influence of VMPr is not statistically significant. Figure 8 shows the progression, regression 1 takes a PHC-curve-adjustment into consideration.

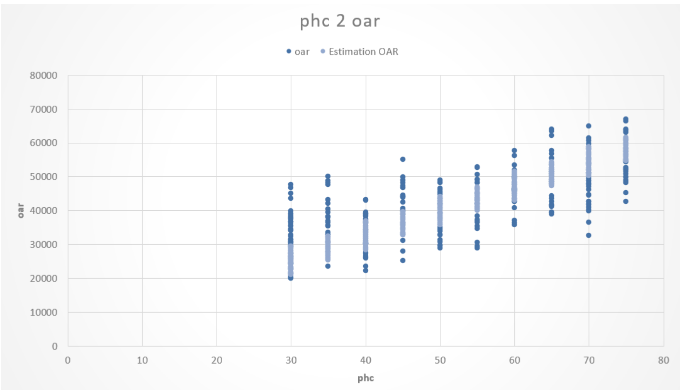

Fig. 8. Regression curve phc to oar, own illustration created with (16) on the basis of data sets extracted from the simulation software

5.2 Regression 2: PSF, PHC and VMPr → PM

During the investigations of regression 2, the impact of the independent variables PSF, PHC and VMPr on the depended variable PM is investigated. Based on 3 degrees of freedom, an ANOVA analysis revealed the F-statistic to be 127.560 in regression 2. This proves its statistical significance. However, the calculation revealed a multiple correlation coefficient to be 0.721 which is below the value of regression 1. Consequently, the adjusted coefficient of determination R^2_{adj} is 0.515. Therefore, the quality of regression 2 is less than the quality of regression 1.

Table T3 shows the impact, the independent variables have, to explain the variance within the model in detail. Like regression 1, regression 2 also shows a high standard error regarding to variable VMPr. However, the analysis of the *t-statistics* revealed the PSF to be less than the marginal value, required to prove its contribution on a significance level of 5 %. Otherwise, the variables PHC and VMPr are able to explain the distribution of the return on sales.

Table 3: Result of the regression analysis, calculated with (16)

variable	coefficient	standard error	t-statistics	p-value
PHC	0,38	0,03	14,86	0,00
PSF	-0,04	0,04	-0,95	0,34
VMPr	-30,99	3,49	-8,89	0,00

Table 3: *Results of regression 2 .*

5.3 Regression 3: PSF, PHC and OAR → MFP

In order to elaborate a model which is applicable and able to fulfil the needs, specified in section 1, a third regression is necessary. This regression is an inverse transformation of OAR on to MFP. Consequently, the third regression facilitates the calculation of a valid MFP on the basis of OAR. Next to OAR, PSF and PHC are independent variables in regression 3.

The calculation of regression 3 revealed a multiple correlation coefficient to be 0.957 and an adjusted coefficient of determination of 0.9125. The standard error is 67.92. As a result of an ANOVA analysis, the F-statistic is 1280.423 which proves the statistical significance of regression 3.

Table 4: Result of the regression analysis, calculated with (16)

variable	coefficient	standard error	t-statistics	p-value
PSF	0.48	0.23	2.13	0.03
PHC	-0.89	0.36	-2.45	0.02
OAR	0.01	0.00	12.70	0.00

Table 4: *Results of regression 3 .*

In table T4 the impact of the independent variables on the quality of the regression is described. The calculation of a *t-statistic* revealed the variables PSF, PHC and OAR to make a significant contribution to the statistical quality of the model. Therefore, regression

3 serves as a basis to find a prediction function in order to calculate the critical MFP. A critical MFP is defined as MFP where software producers have no earnings and no losses.

In order to determine the critical MFP, the minimum turnover has to be calculated at first. In order to do so, the earning-functions ($e := OAR \rightarrow OAR - (OAR \cdot 0.3347 + 22.600)$) point of intersection with the x-axis has to be calculated on the basis of the above mentioned explanations. The axial section is at 33969.63 €.

As a consequence, the critical MFP could be predicted by using the formula $est_{MFP} := (PSF, PHC) \rightarrow 0,4781PSF - 0,891PHC + 207,21$. The figure shows an increasing PSF to require a higher MFP (except for the standard error, mentioned before). Furthermore, a decreasing PHC requires a higher MFP in order to avoid financial losses.

6 Conclusion

In practice, software producers are forced to agree on fixed-price contracts due to economic reasons. In order to calculate a valid fixed-price, software producers need to know the minimum price required. A model to calculate the minimum fixed-price can be restricted to three independent variables. These are the *probability of processing cycles* (PSF), the *potential of cross-subsidisation* (PHC) and the *minimum fixed-prices* (MFP, resp. VMPr).

During a simulation 356 scenarios were analysed which differ in the amount of PSF, PHC and MFP/VMPr. The basis of this simulation-study is the model of a simple software development process. The personal capacity was not changed during the simulation. On the basis of an optimistic assumption, the lowest possible minimum base-price was calculated by assuming a maximum productivity. This minimum base-price was deceeded by a percentage (VMPr, varied in the interval [0%; 50%]) for each scenario before its simulation. This leads to the MFP which was used in order to parametrize the simulation model that belongs to a scenario. During the simulation the amount of accounts receivables was calculated in an optimistic manner for a software producer. This amount of accounts receivables which does not conform to the turnover absolutely (payment-effective), is required by the software producer to cover his whole costs and to realize his earnings. Investigations on income statements of different software producers in Germany show the percentage of fixed costs (except personal costs) of turnover to be about 33.47 %. Therefore, next to the amount of accounts receivables, the return on sales could be considered as dependent variable.

The calculations of three regressions show PHC, PSF and MFP/VMPr to have a high impact on OAR, PCb or PM. The investigation confirms that software producers have to obtain a minimum turnover in order to avoid financial losses. Software producers whose production configurations match those of the model, described in this paper, require a turnover of 33969.63 € at least. Combining this with the results from regression 3, leads to a function, applicable to predict the critical MFP $est_{MFP} := (PSF, PHC) \rightarrow 0,4781PSF - 0,891PHC + 207,21$. The prediction function est_{MFP} enables software producers to determine the required MFP which leads to no earnings and no losses (except the described standard error of 67.92).

In order to derive a simple model which is applicable in practice, a linear regression was calculated. In further investigations it might be tested if other forms of approximations lead to results which are more precise. Furthermore, the time preference of payments (in this case, receivables) might be considered in an enhanced model, in order to determine the value of investments in price reductions by the use of classical methods of investment appraisal.

REFERENCES

[1] Alain Abran, Jean-Marc Desharnais, Mohammad Zarour, and Onur Demirors. Productivity-Based Software Estimation Models and Process Improvement: an Empirical Study. *International Journal on Advances in Software*, 8(1 & 2):103–114, 2015. IARIA.

[2] Oktay Adalier, Aybars Uğur, Serdar Korukoğlu, and Kadir Ertaş. A New Regression Based Software Cost Estimation Model Using Power Values. In Hujun Yin, Peter Tino, Emilio Corchado, Will Byrne, and Xin Yao, editors, *Intelligent Data Engineering and Automated Learning - IDEAL 2007*, volume 4881, pages 326–334. Springer, Berlin, Heidelberg, 2007. DOI: 10.1007/978-3-540-77226-2_34.

[3] Allan J. Albrecht and John E. Gaffney. Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. *IEEE Transactions on Software Engineering*, SE-9(6):639–648, November 1983. DOI: 10.1109/TSE.1983.235271.

[4] Gibeon Soares de Aquino Júnior. *Value-Based Productivity Measurement in Software Development Projects.* Phd thesis, Universidad Federal de Pernambuco, July 2010. Available at *Universidade Federal De Pernambuco*: `http://repositorio.ufpe.br/bitstream/handle/123456789/1768/arquivo3204_1.pdf`, accessed 2017-10-16.

[5] Ove Armbrust, Thomas Berlage, Thomas Hanne, Patrick Lang, Jürgen Münch, Holger Neu, Stefan Nickel, Ioana Rus, Alex Sarishvili, Sascha van Stockum, and Andreas Wirsen. Simulation-based Software Process Modeling and Evaluation. In S K Chang, editor, *Handbook of Software Engineering and Knowledge Engineering*, pages 333–364. World Scientific Publishing Company, 2005. DOI: 10.1142/9789812775245_0012.

[6] John W. Bailey and Victor R. Basili. A meta-model for software development resource expenditures. In *Proceedings of the 5th International Conference on Software Engineering*, pages 107–116. IEEE Press, 1981. Reprint, CH1627.

[7] Alexander Baumeister and Markus Ilg. Entwicklungsbegleitende Kalkulation von Softwareprojekten nach dem Unified Process. *Wirtschaftsinformatik*, 46(3):188–195, June 2004. Berlin, Heidelberg: Springer-Verlag (=WI – Schwerpunktaufsatz). DOI: 10.1007/BF03250936.

[8] Daniela Becker. Analyse der Herausforderungen in der Gestaltung von IT-Verträgen in der agilen Softwareentwicklung. Ausarbeitung, LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN, 2016. Software and Computational Systems Lab, Juristisches IT-Projektmanagement Dr. Frank Sarre. Available at *Software and Computational Systems Lab, Institute of Informatics, Faculty of Mathematics, Informatics, and Statistics of Ludwig-Maximilians-Universität München (LMU Munich)*: `https://www.sosy-lab.org/Teaching/2016-WS-JurPM/Agile-Softwareentwicklung%20(Ausarbeitung).pdf`, accessed 2018-06-18.

[9] Michel Benaroch. Managing Information Technology Investment Risk: A Real Options Perspective. *Journal of Management Information Systems*, 19(2):43–84, January 2002. Taylor & Francis Group. DOI: 10.1080/07421222.2002.11045726.

[10] Tanmay Bhowmik, Nan Niu, Wentao Wang, Jing-Ru C Cheng, Ling Li, and Xiongfei Cao. Optimal Group Size for Software Change Tasks: A Social Information Foraging Perspective. *IEEE Transactions on Cybernetics*, 46(8):1784–1795, August 2016. DOI: 10.1109/TCYB.2015.2420316.

[11] Joseph D Blackburn, Gary D Scudder, and Luk N Van Wassenhove. Improving Speed and Productivity of Software Development: A Global Survey of Software Developers.

IEEE Transactions on Software Engineering, 22(12):875–885, December 1996. DOI: 10.1109/32.553636.

[12] Barry W. Boehm. *SOFTWARE ENGINEERING ECONOMICS.* Prentice-Hall Advances in Computing Science and Technology Series). Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1981. ISBN: 0-13-822122-7.

[13] Barry W Boehm. A Spiral Model of Software Development and Enhancement. *Computer*, 21(5):61–72, May 1988. IEEE. DOI: 10.1109/2.59.

[14] Frederick P. Jr. Brooks. *The Mythical Man-Month. Essay on Software Engineering. Anniversary edition with four new chapters.* Number 20th Anniversary Edition. Addison-Wesley Longman, Inc., Boston, San Francisco, New York and others, 1995. ISBN: 0-201-83595-9.

[15] Hans Ulrich Buhl. Betriebswirtschaftliche Determinanten von Software-Entscheidungen. In Karl Kurbel, editor, *Wirtschaftsinformatik 93: Innovative Anwendungen, Technologie, Integration 8. -10. März 1993, Münster*, pages 154 – 168. Physica-Verlag, Heidelberg, March 1993. DOI: 10.1007/978-3-642-52400-4_13.

[16] Microsoft Corporation. Microsoft© Excel©. Office Home Business 2019, 2019. Version 2212.

[17] Microsoft Corporation. Microsoft© Visio© 2021 MSO. Visio Standard, 2021. Version 2308 Build 16.0.16731.20052, 64 Bit.

[18] Thomas H. Davenport. *Process Innovation. Reengineering Work through Information Technology.* Havard Business School Press, Boston, Massachusetts, 1993. Ernst & Young. ISBN: 0-87584-366-2.

[19] Hyunsook Do and Gregg Rothermel. Using Sensitivity Analysis to Create Simplified Economic Models for Regression Testing. In *ISSTA '08: Proceedings of the 2008 international symposium on Software testing and analysis*, pages 51–61. Association for Computing Machinery (ACM), New York, USA, July 2008. ISBN: 978-1-60558-050-0.

[20] Roy Dornhöfer. Die Abnahme im Werkvertrag. *Juraexamen.info e.V. Online Zeitschrift für Jurastudium, Staatsexamen und Referendariat*, April 2014. Available at *juraexamen.info Die E-Zeitschrift*: `http://www.juraexamen.info/die-abnahme-im-werkvertrag/`, accessed 2018-11-05.

[21] Leonard Farr and Henry J Zagorski. Factors that affect the cost of computer programming a quantitative analysis. Report Vol. II, SYSTEM DEVELOPMENT CORP SANTA MONICA CALIFORNIA, September 1964. REPORT NO. ESD-TDR-64-448.

[22] Klaus Gennen. Auswirkungen hybrider Projektvorgehensmethoden auf den Softwareerstellungsvertrag. In Martin Engstler, Masud Fazal-Baqaie, Eckhart Hanser, Oliver Linssen, Martin Mikusz, and Alexander Volland, editors, *Projektmanagement und Vorgehensmodelle 2016*, Lecture Notes in Informatics (LNI), pages 37 – 48. Gesellschaft für Informatik eV, Bonn, 2016. ISBN 978-3-88579-657-2. URL: `https://dl.gi.de/bitstream/handle/20.500.12116/587/37.pdf`, last access at June 20th, 2018.

[23] Uwe Götze. *Kostenrechnung und Kostenmanagement.* Number 4. Auflage in Springer-Lehrbuch. Springer-Verlag, Berlin, Heidelberg, New York, February 2007. ISBN 978-3-540-32715-8.

[24] Peter Green and Michael Rosemann. Integrated process modeling: An ontological evaluation. *Information Systems*, 25(2):73–87, April 2000. Pergamon. Elsevier Science Ltd.

[25] Erich Gutenberg. *Grundlagen der Betriebswirtschaftslehre. Erster Band. Die Produktion.* Number 24. Auflage in Enzyklopädie der Rechts- und Staatswissenschaft

Abteilung Staatswissenschaft. Springer - Verlag Berlin Heidelberg New York, Berlin, Heidelberg, New York, 1983. ISBN: 978-3-540-05694-2.

[26] Patrick Hamilton. Basteln sie oder produzieren sie schon? In Patrick Hamilton, editor, *Dynaxity: Management von Dynamik und Komplexität im Softwarebau*, pages 11–41. Springer-Verlag, Berlin, Heidelberg, 2007. ISBN-10: 3-540-31743-0. DOI: 10.1007/978-3-540-31744-9_2.

[27] Michael Hammer and James Champy. *Reengineering the Corporation. A Manifesto for Business Revolution.* HarperBusiness A Division of HarperCollins Publishers, New York, USA, 1993. ISBN: 0-88730-687-X.

[28] Marta Indulska, Peter Green, Jan Recker, and Michael Rosemann. Business Process Modeling: Perceived Benefits. In Alberto H. F. Laender, Silvana Castano, Umeshwar Dayal, Fabio Casati, and José Palazzo M. Oliveira, editors, *Conceptual Modeling – ER 2009. 28th International Conference on Conceptual Modeling, Gramado, Brazil, November 9-12, 2009, Proceedings*, volume 5829 of *Lecture Notes in Computer Science*, pages 458–471. Springer-Verlag, Berlin, Heidelberg, 2009. DOI: 10.1007/978-3-642-04840-1_34.

[29] Magne Jørgensen. Regression Models of Software Development Effort Estimation Accuracy and Bias. *Empirical Software Engineering*, 9(4):297–314, December 2004. Springer, Kluwer Academic Publishers. DOI: 10.1023/B:EMSE.0000039881.57613.cb.

[30] Noureddine Kerzazi. Localizing Inconsistencies into Software Process Models at a Conceptual Level. In Álvaro Rocha, Hojjat Adeli, Luís Paulo Reis, and Costanzo Sandra, editors, *New Knowledge in Information Systems and Technologies*, volume 1 of *Advances in Intelligent Systems and Computing (AISC, volume 930)*, pages 775 – 788. Springer Nature Switzerland AG, Cham, Switzerland, March 2019. DOI: 10.1007/978-3-030-16181-1_73. Print ISBN 978-3-030-16180-4. Online ISBN 978-3-030-16181-1.

[31] Wolfgang Kilger, Jochen R. Pampel, and Kurt Vikas. *Flexible Plankostenrechnung und Deckungsbeitragsrechnung.* Number 13. Auflage. Gabler Verlag, Springer Fachmedien Wiesbaden, Wiesbaden, 2012. DOI: 10.1007/978-3-8349-3758-2.

[32] Hans-Bernd Kittlaus and Peter N Clough. Chapter 5: The Elements of Software Pricing. In Hans-Bernd Kittlaus and Pater N. Clough, editors, *Software Product Management and Pricing. Key Success Factors for Software Organizations*, pages 117–194. Springer, Berlin, Heidelberg, 2009. DOI: 10.1007/978-3-540-76987-3_5.

[33] Gerhard Knolmayer. Das Brooks' sche Gesetz: Überlegungen zum Zusammenhang zwischen Teamgröße und Projektdauer bei der Software-Produktion. Working Paper 190, Institut für Betriebswirtschaftslehre der Universität Kiel, Kiel, Germany, 1987.

[34] David Kuhlen. *Kostensenkungspotenzialfunktion der Softwareproduktion.* Phd thesis, Christian-Albrechts-Universität zu Kiel, Kiel, Germany, 2019.

[35] David Kuhlen and Andreas Speck. Economic output under the conditions of social freedom in software development. *International Journal of Software Engineering & Applications (IJSEA)*, 7(6):17 – 31, November 2016. DOI: 10.5121/ijsea.2016.7602.

[36] David Kuhlen and Andreas Speck. Efficient Key Performance Indicator Calculation. *International Journal of Software Engineering & Applications (IJSEA)*, 8(5):1–15, September 2017. DOI: 10.5121/ijsea.2017.8501.

[37] David Kuhlen and Andreas Speck. The way of designing a simulation software in order to evaluate the economic performance in software development. In *Proceedings of The 8th International Conference on Computer Modeling and Simulation. ICCMS 2017. Canberra, Australia January 20-23, 2017*, International Conference Proceedings Series by ACM, pages 109 – 113. Association for Computing Machinery (ACM), New York, USA, January 2017. DOI: 10.1145/3036331.3036347.

[38] C. Lucas, Henry, J. Walton, Eric, and Michael J. Ginzberg. Implementing Packaged Software. *MIS Quarterly*, 12(4):537 – 549, December 1988. ACM Categories: H.2.4, H.4.m, J.I.

[39] Samer I Mohamed. Software Productivity from an Industrial Perspective. In Haitham Hamza and Azza Kamal, editors, *Conference Proceedings of the 1st Africa and Middle East Conference on Software Engineering (AMECSE 2014)*, pages 95 – 101, October 2014.

[40] Arno Müller, Lars von Thienen, and Hinrich Schröder. IT-Controlling: So messen Sie den Beitrag der Informationstechnologie zum Unternehmenserfolg. *Arbeitspapiere der Nordakademie*, (Nr. 2004-02), November 2004. ISSN: 1860-0360.

[41] Johannes Müller. Software-Ökonomik in Hinblick auf Software-Produktlinien. State of the Art. *SSRN*, March 2011. DOI: 10.2139/ssrn.1808883.

[42] Marc Prud'homme, Dimitri Sanga, and Kam Yu. A computer software price index using scanner data. *Canadian Journal of Economics/Revue canadienne d'économique*, 38(3):999–1017, August 2005. Wiley Online Library. DOI: 10.1111/j.0008-4085.2005.00313.x.

[43] Mirjana Radonjic-Simic, Frank Wolff, and Dennis Pfisterer. Analyzing an Ecosystem for Complex Consumer Services. In Yoshinori Hara and Dimitris Karagiannis, editors, *Serviceology for Services. 5th International Conference, ICServ 2017, Vienna, Austria, July 12-14, 2017, Proceedings*, volume 10371 of *Lecture Notes in Computer Science (LNPSE)*, pages 46 – 52. Springer International Publishing AG, Cham, Switzerland, 2017. DOI: 10.1007/978-3-319-61240-9_5.

[44] Hussain Saleem and S. M. Aqil Burney. Imposing Software Traceability and Configuration Management for Change Tolerance in Software Production. *IJCSNS International Journal of Computer Science and Network Security*, 19(1):145–154, January 2019.

[45] Hinrich Schröder, Ralf Kesten, and Thomas Hartwich. Produktorientierte IT-Leistungsverrechnung bei der K+S Gruppe. *HMD - Praxis der Wirtschaftsinformatik*, 44(254):50 – 60, April 2007. Springer. DOI: 10.1007/BF03340267.

[46] Andreas Speck. Reusable Industrial Control Systems. *IEEE Transactions on Industrial Electronics*, 50(3):412–418, June 2003. DOI: 10.1109/TIE.2003.812274.

[47] Andreas Speck, Sven Feja, Sören Witt, Elke Pulvermüller, and Marcel Schulz. Formalizing Business Process Specifications. *Computer Science and Information Systems (ComSIS)*, 8(2):427–446, May 2011. DOI: 10.2298/CSIS110111015S.

[48] Moutasm Tamimi, Fatimah Alghamdi, and Ahid Yaseen. A Systematic Snapshot Review of Custom-Made Software Enterprises from the Development Perspectives. *International Journal of Information Systems Management Research & Development (IJISMRD)*, 9(1):1 – 24, February 2019. ISSN (P): 2250-236X. ISSN (E): 2319-4480. Copyright TJPRC Pvt. Ltd.

[49] Adam Trendowicz and Jürgen Münch. Factors Influencing Software Development Productivity-State-of-the-Art and Industrial Experiences. In Marvin V. Zelkowitz, editor, *Advances in Computers*, volume 77, chapter Chapter 6, pages 185–241. Academic Press, Elsevier Inc., London, San Diego, Cambridge, 2009. DOI: 10.1016/S0065-2458(09)01206-6.

[50] The TeX Users Group (TUG). TeX Live, 2024. Version 3.141592653-2.6-1.40.26.

[51] Falk Uebernickel, Jochen Scheeg, and Walter Brenner. Produkt- und serviceorientiertes IT-Controlling am Fallbeispiel T-Systems ActiveBilling. *HMD – Praxis der Wirtschaftsinformatik*, 44(254):39 – 49, April 2007. DOI: 10.1007/BF03340266.

[52] Kostas Vergidis, Ashutosh Tiwari, and Basim Majeed. Business Process Analysis and Optimization: Beyond Reengineering. *IEEE Transactions on Systems, Man, and*

Cybernetics, Part C (Applications and Reviews), 38(1):69–82, January 2008. DOI: 10.1109/TSMCC.2007.905812.

[53] Claude E. Walston and Charles P. Felix. A method of programming measurement and estimation. *IBM Systems Journal*, 16(1):54–73, 1977. IEEE. DOI: 10.1147/sj.161.0054.

[54] Matthias Wolf. Sicherstellung von Qualität in der agilen Softwareentwicklung. Diplomarbeit, Ludwig-Maximilians-Universität (LMU) München, September 2009. Institut für Informatik Lehr- und Forschungseinheit Programmierung und Softwaretechnik. Available at *LMU Munich*: `http://tmi.pst.ifi.lmu.de/thesis_results/0000/0040/DA-WOLFM-QUALITAET-AGIL.pdf`, accessed 2015-04-03.