Received: 9 December 2020 | Revised: 25 February 2021 | Accepted: 28 February 2021
DOI: 10.1002/mde.3336

RESEARCH ARTICLE

WILEY

Valuation and pricing of software licenses to support supplier–buyer negotiations: A case study in the automotive industry

Frank Bodendorf[1] | Manuel Lutz[2] | Jörg Franke[1]

[1]Institute for Factory Automation and Production Systems, Friedrich-Alexander-University of Erlangen-Nuremberg, Erlangen, Germany

[2]TUM School of Management, Technical University of Munich, Munich, Germany

Correspondence
Frank Bodendorf, Institute for Factory Automation and Production Systems, Friedrich-Alexander-University of Erlangen-Nuremberg, Egerlandstraße 7-9, 91058. Erlangen, Germany.
Email: frank.bodendorf@faps.fau.de

Licensed standard software can be replicated millions of times at near zero cost for the supplier and poses immense challenges for purchasing and cost managers. A model is introduced to evaluate the appropriateness of software prices and thus to support the negotiation and decision process regarding suppliers. The model determines the "value to customer" of a software product using conjoint analysis and converts that value into a theoretical license price. In a case study coming from a large Original Equipment Manufacturer (OEM) it is shown that the developed conjoint model can successfully be used in practice and suggests realistic purchase prices.

1 | INTRODUCTION

Digitalization is one of the main drivers of globalization and economic growth. For few sectors this applies more than for the manufacturing industry, characterized by a competitive market with an abundance of specialized players and a deeply hierarchical value chain. Besides, the recent advance of customer-centric market philosophies and resulting increased customer expectations have imposed strong pressure on companies to install and maintain high-quality product standards (Hansen & Bush, 1999) and the automotive industry in particular depends heavily on its promise of premium products. Another side effect of this development is the necessity to offer a broad portfolio of customizable products. Although this may serve as a selling proposition, it also shortens product life cycles, requiring companies to accelerate their processes. Due to the increasing digitalization of the automobile, established Original Equipment Manufacturers (OEMs) in the automotive industry and the associated supplier industry are currently facing serious and complex challenges (Llopis-Albert et al., 2020). The digital transformation is changing the automotive industry sustainably through intelligent information and communication technology as well as autonomous, interacting, and networked systems with intelligent sensors (Bertozzi et al., 2000). As a result, the automobile is radically changing from a conventional means of transport to a fully networked software platform, enabling car manufacturers to offer a comprehensive portfolio of individual, intelligent, and innovative services. However, this development inevitably leads to a fundamentally changing competitive environment with new market players (Hylving, 2015). Software products are becoming a central and differentiating factor and a basis for new, innovative business models. The enormous importance of software thus leads to a certain dependence of automobile manufacturers on software suppliers and innovative software companies. This dependency becomes a fundamental problem, especially in the evaluation of software products to be procured. The development costs for a software component can only be observed or evaluated to a limited extent by the automobile manufacturers as customers. The software companies hold an information advantage due to a higher level of experience and detailed knowledge of internal processes (Steinle et al., 2014). As a result, a business relationship develops which is characterized by a fundamental asymmetry of information and a high degree of distrust between the actors involved. In addition, customers are no longer prepared to pay significantly higher prices for technical innovations. They are increasingly price-sensitive but expect comprehensive standard equipment (Kübler et al., 2018). As a result of these trends, OEMs today are confronted with ever higher spending on automotive software. In order to identify cost potentials and not to jeopardize profitability, it is now up to the automotive manufacturers' purchasing department to take action. Over the last decades, however, this division has focused its processes on the procurement of electronics hardware and material components (Cousins et al., 2006). In recent years, modular strategies and clearly defined purchasing processes have ensured high cost efficiency for simple purchased and standard parts (Handfield et al., 2002). However, these processes are only of limited use for the procurement of software due to their specific product characteristics (Koppius, 1999). The situation is further complicated by the fact that the pricing strategies of software providers are constantly changing (Lehmann & Buxmann, 2009). The methods previously used to analyze purchase prices in practice cannot be easily applied to software licenses. Suitable approaches must make it possible to quantify software license models in monetary terms and to evaluate the appropriateness of their offer prices. In this paper approaches and methods are investigated that can be used to evaluate software licenses with respect to their price adequacy. First, the current state of research in literature on the valuation of software license prices is presented in Section 2. In Section 3, a model based on conjoint analysis is introduced that enables the monetary valuation of software licenses out of a customer perspective. The method is applied in a real-world case study at a large automotive manufacturer. Section 4 discusses limitations and implications for purchasing and supply management. Section 5 concludes the research project.

2 | LITERATURE REVIEW

2.1 | Review method

The research streams coming from Webster and Watson (2002) and Brocke et al. (2009) offer two approaches to literature review. Whereas the first gives recommendations on how to conduct a structured literature search process, the second discusses guidelines for structuring and classifying the results. The literature review in this section is based on both approaches mentioned above. A modified process model describes the different phases of the literature search procedure (see Figure 1(a)). In phase (I) the research focus is determined. The objectives of the literature search are defined and keywords for the database search are determined. Phase (II) of the search describes the operational search process for relevant publications in various databases. In phase (III) the identified literature is categorically structured in a concept matrix. Finally, in phase (IV) the literature is evaluated with respect to the current state of research.

The literature review aims to identify methods, techniques, and approaches that are applied for the evaluation of software license prices. In order to proceed as efficiently as possible, a search string is defined, which contains relevant words as well as their synonyms. Because the search is conducted in German and English language bibliographies, keywords in both languages are used. For example an English search term is (software OR digital good*) AND licen* AND (pric* OR cost* OR fee* OR charg* OR rate* OR royalt*).

The procedure of the literature review is based on the methodology of Brocke et al. (2009) and is shown in Figure 1(b). In the first step suitable scientific databases from the fields of economics and computer science are identified. A comprehensive search is performed using the databases AIS Electronic Library (AISel), EconBiz, EBSCO Business Source Complete, Emerald Insight, Informs, ScienceDirect, Scopus, SpringerLink, Web of Science and WISO. After collecting potentially interesting publications from all databases (step 2), duplicates are identified and eliminated (step 3). In the next step titles and abstracts of the literature found are analyzed to exclude non-relevant publications. For the remaining publications a full text analysis is performed. In order to enhance the completeness of the considered literature, additional relevant literature is identified by backward and forward searches. Furthermore, the search engine Google Scholar is used to find more potentially relevant literature. In total, the contents of 49 scientific articles, books, and conference papers are analyzed in detail.

By a concept matrix it can be shown which concepts are pursued in which papers. This enables an effective categorization of the articles as well as an identification of similar concepts within the respective research area. A complete overview of all publications and concepts can be found in the Appendix.

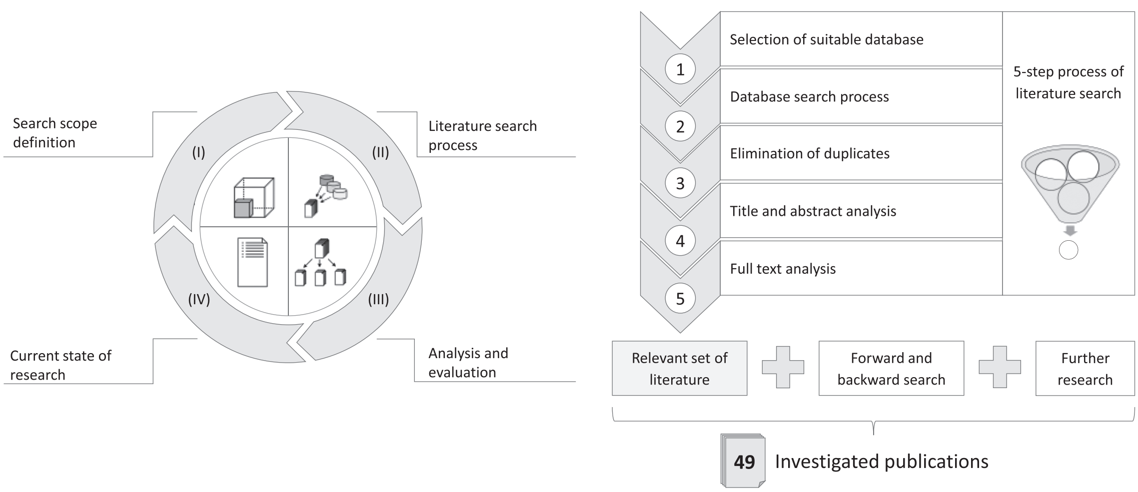

FIGURE 1 Literature review procedure (a) and process steps (b)

2.2 | Review results

In the following the most significant concepts are briefly presented based on the most important publications.

Tam et al. (2020) find out that customer involvement is one of the main factors contributing to the success of ongoing agile software development projects. Tsunoda et al. (2012) deal with the question of how a quoted price can be validated from the customer's perspective. However, only a few influencing factors on the price are examined and no model is provided to validate this. Furthermore, the context of the paper refers to individual software and not to software licenses.

Keshta (2017) and Bilgaiyan et al. (2017) give an overview of important software cost estimation methods. Pospieszny et al. (2018) and Dragicevic et al. (2017) expand traditional parametric approaches by effective and practical machine learning approaches to estimate effort and duration of software development projects. Nestler (2013) provides concepts for the valuation of intangible assets based on the principles of the Institute of Public Auditors in Germany (IDW). She proposes to determine the license fee by means of a profit split. However, the determined license value can only be seen as a guideline, as it is not grounded on a sound methodology. Nestler is also working on the financially oriented evaluation of (individual) software. She emphasizes that there is no recognized standard for the valuation of software.

Baumeister and Ilg (2004) deal with cost forecasts for software development projects that are carried out according to the "Unified Process". Among other things, they use phase-related, similarity-determining cost-driver rates. For similarity analysis, the authors consider product, computer, personnel, and project influencing variables. The authors argue that the selection of influencing variables for a similarity analysis should always be handled situation-specifically. A similarity-based cost estimation for software is an interesting approach to determine the license price. However, Baumeister and Ilg (2004) do not provide a concrete model for this and the influencing variables they mention are not known in license purchasing.

Sequeira and Lopes (2015) estimate the costs of a project on the basis of a work breakdown structure. In the first step, this structure is converted into a "responsibility assignment matrix". Subsequently, the time required for the respective work packages is estimated and charged at the corresponding hourly wage rates. Dewi and Subriadi (2017) further develop the established Function Point method (Symons, 1988) by a modification of complexity factors in order to estimate the software development effort. Closely related to the Function Point method, the algorithmic cost model Constructive Cost Model (COCOMO) is also used in software development for cost or effort estimation. Miller et al. (2018) demonstrate the applicability of the COCOMO in practice.

Singh et al. (2018) estimate the effort and costs of software development from the manufacturer's perspective based on a parametric algorithm using historical information from past projects. The information comprises the lines of code, the methodology used, and the actual effort required. Following the ideas of Pospieszny et al. (2018), Venkataiah et al. (2019) propose a non-parametric approach using an Artificial Neural Network in order to improve the accuracy of process cost estimation which is improving the quality of the software product. However, a straight application of the methodologies in the context of software license prices is not viable due to the lack of input data.

Lehmann and Buxmann (2009) examine the pricing strategies of software vendors and identify various parameters that usually make up the combination of software vendors' pricing models. They conclude that demand-based or value-based pricing should be preferred to cost-based pricing. Harmon et al. (2009) take up the idea of value-based pricing and develop a framework for a corresponding model. Rohitratana and Altmann (2012) present an agent-based simulation system that allows modeling the interactions between software buyers and vendors in a software market under different pricing schemes. Zhang (2020) takes a game theoretical perspective focusing on the analysis of pricing strategies from a customer's perspective. He finds out that it is not always optimal for the software firm to discriminate between its old users and the new customers. Jia et al. (2018) expand the horizon of observation and examine and analyze the profitability of a selling and a leasing model by considering both software upgrades and different price discrimination strategies. The above mentioned publications deal with the pricing of software and allow deep insights into software developers' pricing strategies, but do not give any indication how a price should be evaluated from the buyer's point of view.

2.3 | Lessons learned

None of the approaches mentioned in Section 2.1 deals with the price evaluation of software licenses in procurement. Only Tsunoda et al. (2012) discuss price plausibility checks for individual software from the buyer's perspective (Tsunoda et al., 2012). Therefore, the literature review does not provide a satisfactory solution to the problem addressed by the topic of this paper. It turns out that software licenses are usually priced on a value basis (Töytäri et al., 2015). Therefore, a plausibility check of the license price based on the value of the software for the customer seems to be the right approach. However, an appropriate model that determines the purchase price of software on a value basis cannot be found in the literature. Section 3 introduces such a model, that uses the methodology of conjoint analysis, and illustrates the model application by a case study coming from automotive manufacturing.

3 | CONJOINT ANALYSIS MODEL

3.1 | Theoretical background

The conjoint analysis (CA), known from marketing, is applicable in pricing decisions for new products, taking into account the difficulty of product cost structure determination. For this purpose, one idea is to consider the individual software functions, attach partial utility values, and derive total utility values of system variants from these values (Carroll & Green, 1995). A different way is to compare the utility values of several software systems or products in order to derive the relative willingness to pay for the software product from the known willingness to pay of the other products. In the following, the procedure for such a conjoint analysis is outlined and then carried out in detail in Section 3.2.

In principle, conjoint analysis can be divided into traditional and modern methods. Traditional methods aim at creating a preference ranking over all objects, for example, different software products or different sub-functions of a software system, by querying preference judgments directly. This is done either by a ranking (order of precedence) or a rating (point values) (Kalish & Nelson, 1991).

Hybrid methods combine the conjoint approach with a scoring model (Green & Krieger, 1996). Thus, the adaptive conjoint analysis (ACA) consists of two parts. In the compositional part, all properties and their characteristics are evaluated by the participants. This serves to determine the individual importance of the characteristics and to estimate the benefit contributions. In the subsequent decomposition part, pair comparisons of the so-called stimuli are carried out, thereby adjusting the benefit estimates from the first part and finally, selection probabilities are determined (Green et al., 1991). The ACA has the advantage that the survey design also allows the evaluation of products with a larger number of characteristics (up to 30) and corresponding respective characteristics (up to 9), compared to traditional CA. With the selection-based conjoint analysis (SBCA), no scoring is done, but a set of alternatives is presented, and a concrete selection decision is then requested. With choice-based conjoint analysis (CBCA), the respondents select one stimulus or no stimulus at all from a set of stimuli. The preferences are then derived directly from the selection decisions made (Elrod et al., 1992). This is closer to reality than traditional conjoint analysis (TCA). However, the selection decisions made in CBCA contain only binary 0/1 information; that is, the data is scaled nominally. This more realistic query of preferences therefore leads to a loss of information compared to the TCA. It is also difficult to establish a utility function with partial utility values, which is usually done by a regression approach. This means that other methods are needed to estimate the preference function. Although the data obtained in TCA can be analyzed relatively easily by regression analysis, CBCA requires more complex procedures such as the maximum likelihood method or other iterative optimization. Even though the CBCA, in contrast to the TCA, can calculate additional selection probabilities, which can be used, for example, for prognosis scenarios (Natter & Feurstein, 2002), the TCA is preferred to the CBCA due to the less complex concept and the simultaneous evaluation of software benefits. In this paper, the applicability of this TCA approach is investigated in a case study-based research. The research results are discussed in the light of realistic cost estimation of software licenses.

3.2 | Case Study

The case study comes from a large Bavarian automotive manufacturer. A conjoint analysis to determine a software price is shown for a real-world voice control system with the fictitious name "Speech" that is used in the vehicle. Speech processing is generally divided into an onboard and offboard component. In contrast to the onboard component, the offboard component uploads the speech input to a remote server. This increases the range of possible commands and functions. In the use case investigated separate prices are determined, among others, for the onboard and the offboard software component of Speech. The procedure for the conjoint analysis follows the standard structure, for example, given in Gustafsson et al. (2007). All statistical analyses are performed using IBM SPSS Statistics 23.0 software (Arbuckle, 2011).

3.2.1 | Definition of properties

In order to determine partial utility values for specific properties, the characteristics to be queried must first be defined. In the use case, the partial utility values are calculated for specific software functions and the total utility value is calculated by summing them up. A fictitious digital services package is formed to query preferential opinions. Such service packages are offered in similar forms by various car manufacturers. In the following, the characteristics of the package versions are: price (100€, 140€, 180€), voice control (not available, only onboard, onboard & offboard), parking information (not available, available) and remote services (not available, available).

3.2.2 | Experimental design

When selecting the survey design (also called experimental design), decisions have to be made regarding the number of stimuli to be evaluated. In the full-profile method, a stimulus consists of a specific value combination of all characteristics of the product to be evaluated. In contrast, the two-factor method (also known as trade-off analysis) only takes pairs of characteristics into account. For each possible pair a trade-off matrix is created, which contains all conceivable combinations of the values of both characteristics. Each cell represents one stimulus. The two-factor method has the advantage that it is easier for the test person to rate because only two characteristics have to be looked at simultaneously. The full-profile method, on the other hand, is classified as much more realistic, so that it is used more often in practice (Pullman et al., 1999). For this reason, the full-profile method is selected for this use case. This results in $3 \times 3 \times 2 \times 2 = 36$ different stimuli. An example for a stimulus is (price = 100€ | voice control = only onboard | parking information = available | remote services = available).

3.2.3 | Presentation and evaluation of stimuli

The test persons place the stimuli in an order of preference according to their ideas of benefit. This can be done in different ways. In general, a distinction is made between metric and non-metric approaches. In a ranking, the stimuli to be evaluated are put into a preference order based on the perceived benefit. Each stimulus is assigned an ordinally scaled ranking value. Therefore, the ranking represents a non-metric approach.

In our use case of evaluating a digital services package from a buyer's perspective, it makes sense to ask for a purchase probability. The preference labeling sets the extremes to "not at all" to "totally sure". Intermediate levels are explicitly predefined, as this allows the respondents to evaluate more "intuitively". The survey for preference evaluation is carried out using the online software Qualtrics (Qualtrics, 2018). The survey is distributed via an Internet link to 241 employees of an automobile manufacturer and external managers with a business or technical background.

3.2.4 | Determination and aggregation of partial utility values

Based on the collected preference judgments, the partial utility values of all property characteristics are now determined with the help of a conjoint analysis. For each of the feature values, a partial utility value β_{jm} is estimated first (Scholl et al., 2005). The total utility value y_k of a stimulus k results from the aggregation of the partial utility values. Therefore, the preference model comprises two steps. First, a utility function must be specified, which defines the relationship between property values and utility evaluations (partial utilities). In the second step, an aggregation function of the partial utility values is required (Greco et al., 2004). In order to be able to explain the relationship between features and (partial) benefits, three basic utility functions are conceivable (ideal point model, vector model, partial value model) (Darmon & Rouziès, 1994). The ideal point model assumes a benefit-maximum (optimal) property value. If this value is exceeded or undershot, the benefit is reduced. The vector model assumes a linear relationship (positive or negative) between the values of a property characteristic and the benefit of the product. This model is often used for price characteristics. There is usually a negative relationship. The partial value model is a discrete model and determines a specific partial utility for each characteristic value of a property. It is mainly used for qualitative characteristics such as color, shape or material. It is applied in this use case as there are qualitative values, for example, available/not available. For the calculation of the total utility values in the evaluation model, an additive linkage of the partial utility values is preferred (see Formula 1).

$$y_k = \mu + \sum_{j=1}^{J} \sum_{m=1}^{M_j} \beta_{jm} \cdot x_{jmk} \quad (1)$$

y_k estimated total utility value for stimulus k.

β_{jm} partial utility value for characteristic m of property j

$x_{jmk} \begin{cases} 1, \text{ if stimulus } k \text{ is accompanied by the property } j \text{ in the expression } m \\ 0 \text{ otherwise} \end{cases}$

μ constant

J number of properties

M_j number of expressions of property j

The resulting benefit model is thus compensatory. This means that a low benefit of a specific property can be compensated by a higher partial benefit of another property. In multiplicative models, on the other hand, this is not possible, because here a partial utility value of 0 also leads to a total utility value of 0. The benefit of a product can be interpreted here as the net benefit that a customer receives when purchasing the respective package version at the price mentioned in the stimulus. It can be expressed as the difference between package benefit (gross benefit) minus the money to be paid for the respective package (monetary loss).

The data obtained from the survey is examined by means of a metric analysis of variance. For this, a constant term μ is introduced into Formula 1. This constant represents the average evaluation and can be interpreted as a kind of basic utility. The various property values influence this value positively or negatively. The conjoint analysis algorithm determines the partial utility values β_{jm} for the model (see Formula 1) in such a way that the calculated total utility values y_k correspond as closely as possible to the empirical rating values p_k coming from the survey and the rated stimuli. For this purpose, the least-squares estimation is used, which minimizes the sum of the quadratic deviations between empirical and estimated deviations (see Formulas 2 and 3 and Table 1).

$$\text{Find } \beta_{jm} \text{ with } \min_{\beta jm} \sum_{k=1}^{K} (p_k - y_k)^2 \quad (2)$$

β_{jm} see Formula 1

$$K = N \cdot \prod_{j=1}^{J} M_j \quad (3)$$

K number of stimuli ratings N number of probands

The standard error values shown in Table 1 provide an initial indication of the results' quality. The lower the standard error, the more likely it is that the empirical scale evaluations can be represented by the ratings determined by the conjoint analysis. With a range of variation between 0.116 and 0.605, the preference model therefore delivers good results.

Now the total utility values of each stimulus are calculated (see Formula 4). This is shown exemplarily for Stimulus 1 (price = 180€ | voice control = not available | parking information = available | remote services = available) according to Formula 1.

$$y_1 = 4.877 + (-2.294) + (-1.127) + 0.584 + 0.581 = 2.621 \quad (4)$$

The calculated total utility value of 2.621 is very close to the average rating (=2.440) of all 241 probands for this stimulus. So, the results of the utility function seem to be realistic.

For the monetary conversion of the preference ratings, a negative linear relationship between price and resulting benefit is assumed (vector model). In contrast, a partial value model is used for the analysis of the features voice control, remote services, and parking information. As expected, the calculated partial utility values confirm the

TABLE 1 Partial utility values and parameters calculated using the conjoint analysis

		Partial utility	Standard error
Voice control	Not available	−1.127	0.155
	Onboard	−0.038	0.155
	Onboard + offboard	1.165	0.155
Remote services	Not available	−0.581	0.116
	Available	0.581	0.116
Parking information	Available	0.584	0.116
	Not available	−0.584	0.116
Price	100 €	−1.1274	0.336
	140 €	−1.784	0.471
	180 €	−2.294	0.605
(Constant)		4.877	0.486
Coefficient correlation (observation–estimation)		**Value**	
Price coefficient (b)		−0.13	
Pearson correlation		0.989	
Kendall correlation		1.0	

assumed negative linear relationship between price and resulting benefit. This can be expressed by a price coefficient *b*. It describes the utility adjustment in the event of a price change of one Euro and represents the slope of the utility function.

The monetary utility $\Delta p_{j,m}$ of the value *m* of a property *j* is calculated by Formula 5.

$$\Delta p_{j,m} = \frac{\beta_{j,m} - \beta_{j,\ not\ available}}{|b|} \tag{5}$$

$\Delta p_{j,m}$ monetary value for characteristic *m* of property *j*
β_{jm} partial utility value for characteristic *m* of property *j*
b price coefficient

Formula 5 is used to determine the monetary values of the onboard and offboard components as an example (see Formula 6).

$$\Delta p_{voice\ processing,\ \ Onboard\&Offboard} = \frac{1.165-(-1.127)}{|-0.013|} = \frac{2.292}{|-0.013|} = 176.31 \tag{6}$$

This results in a value for full voice control of € 176.31. Of this, € 83.77 is accounted for by onboard and € 92.54 by offboard voice processing. The calculated amounts represent the monetary value of the respective component from the customer's point of view.

3.2.5 | Assessment of model quality

Table 1 lists the standard errors of the calculated partial utility values. This provides a first indication of the quality of the results. The lower the standard error, the more easily the empirical scale evaluations can be represented by the determined ratings. With a fluctuation range between 0.116 and 0.605, the preference model delivers good results. Further measures for assessing the quality of the model are Pearson's correlation coefficient (Pearson-R) and Kendall's Tau (Croux & Dehon, 2010). Pearson-R reflects the correlation between the calculated total utility values and the actual (empirical) ratings. Kendall's Tau ranks the stimuli based on the actual and calculated utility values and calculates the correlation between the resulting ranks. The closer the correlation coefficients are to the maximum value 1, the better the empirical data can be represented by the results of the conjoint analysis. The values of the correlation coefficients indicate a good representativeness. Furthermore, Kendall's Tau is also calculated for the holdout maps. This coefficient is a measure of the validity of the results and is very high at 1.0. Because the model quality is assessed as sufficiently high, the model values can be used as input for the theoretical calculation of a license price.

3.2.6 | Conversion of the utility value into a license price

As depicted in Section 3.2.4, the monetary value of the voice control from the customer's perspective is € 176.31 (onboard and offboard component together). This value corresponds to the gross benefit of voice control and is now converted into an "optimal" price. To perform a mathematical price optimization, it is necessary to set up the profit function. For this purpose, the price sales function must be known. This is derived from the product benefit in the following. The highest possible sales of the voice control q_{max} is limited by the number of vehicles sold. It is assumed that for a product with a positive gross benefit, the maximum sales volume q_{max} is achieved at a price $p = 0$. If the price p is being increased successively, the customer benefit is reduced. At the maximum price p_{max} (=€176.31) the net benefit (=gross benefit – price to be paid) finally takes a value of 0. Formula 7 shows the utility function of the full-featured speech control.

$$N(p) = Benefit + b \cdot p = 2.292 - 0.013\,p \tag{7}$$

$N(p)$ monetary utility as a function of price
Benefit total utility value of the characteristic
b price coefficient
P price in monetary units

The price sales function $q(p)$ is now set up as a function of the price or the resulting benefit (see Formula 8). It is assumed that the relationship is linear. The quotient $\left(\frac{N(p)}{\text{Benefit}}\right)$ is normalized to the value 1, so that $q(p)$ cannot exceed q_{max}.

$$q(p) = q_{max} \cdot \frac{N(p)}{Benefit} \tag{8}$$

It can be seen that an increasing price leads to a decreasing benefit and thus to lower sales. The profit function $G(p)$ can now be determined using the sales function (see Formula 9).

$$\begin{aligned} G(p) &= U - K = p \cdot q(p) - K[q(p)] = p \cdot q(p) - K_f \\ &= p \cdot \left(q_{max} \cdot \frac{Benefit + b \cdot p}{Benefit}\right) - K_f = \frac{q_{max}}{Benefit} \cdot \left(p \cdot Benefit + b \cdot p^2\right) - 1 \end{aligned} \tag{9}$$

U revenue
K total costs
p price
$q(p)$ price sales function
K_f fixed costs of development (first copy costs)

The costs for the duplication of a software license are striving towards 0 and are thus independent of the sales volume. Therefore, only the fixed costs of development K_f are included in the equation. The profit-optimal price p^* is determined by derivation and zeroing of Formula 9 (see Formula 10).

$$G'(p) = \frac{q_{max}}{Benefit} \cdot (Benefit + 2 \cdot b \cdot p) = 0 \tag{10}$$

This results in Formula 11 calculating the optimal price p^*.

$$p^* = \frac{-Benefit}{2b} = \frac{-2.292}{2 \cdot (-0.013)} = 88.15 \tag{11}$$

The optimal price for the voice control system Speech from the customer's point of view is therefore € 88.15.

By setting the retail price for Speech at € 88.15, the OEM could maximize the unit profit and sales volume.

However, the value determined does not yet represent a purchase price from the OEM's point of view. A profit split, should therefore be carried out between OEM and supplier. The purchase price is defined as a percentage share of the profit generated by the voice control software, with the OEM selling the software to the car buyers. However, the question of an appropriate profit share cannot be answered with a single percentage for all products, suppliers, industries and market situations. In practice, the so-called "Knoppe formula" or the "25 Per Cent Rule" are often used for the valuation of licenses (Hoe & Diltz, 2012). These are based on the idea that the future profits related to the license are to be divided according to the risk profiles of the licensor and licensee. In practice, a share of 25% of the attributable profit has become established as a benchmark. This is to be paid by the licensee to the licensor. In the OEM-supplier scenario of this paper, a risk profile analysis is not useful for determining the profit split. Typically, risks such as market, sales, product liability or warranty risks for the contractual partners have to be evaluated objectively. This requires detailed information about the economic situation and the business strategies of both parties. For this reason, the Knoppe formula cannot be applied one-to-one to software licenses in the automotive sector. However, the value of 25% serves as a point of reference and is defined as the maximum achievable percentage profit share for the supplier. The minimum profit split is set to 0%. To determine an appropriate share level between the two poles mentioned, a scoring model is used according to the recommendations of Simon and Fassnacht (2016). For this purpose, eight factors are selected and evaluated according to the recommendations of Hartmann (2010) for the evaluation of technologies on a five-point scale from 0 to 4 (see Table 2).

The criteria can be divided into three categories: market and supplier-related, product-related and cooperation-related. The first group of criteria includes the factors market structure, market share and flagship product. Market structure usually has a significant influence on the level of profit margin. If a supplier has a monopoly, he can demand significantly higher profit shares than in a market where there is strong competition. For software products a natural monopoly can emerge in the software industry under certain conditions. Therefore, the market structure plays an important role in determining the profit split. A likewise significant influencing variable is the market share. Often there is only one dominant supplier on software markets. As a rule, this supplier benefits from strongly pronounced economies of scale and falling unit costs. This should have a positive effect on the purchase price from the OEM's point of view. In the literature, a supplier is considered dominant if it can claim 40% of the sales volume in the market under consideration (Specht, 2013). Also, software companies often have so-called "flagship products" in their portfolio, which as "cash cows" contribute to the majority of the company's profit. These are characterized by the fact that they are responsible for a large part of the revenue and are positioned in a strongly growing market. It should therefore be checked, for example using the growth–share matrix (BCG) (Schawel & Billing, 2018), whether the product is one of those. Flagship products are already highly profitable, which translates into a lower profit share. The

TABLE 2 Profit split criteria for software licenses

Max. 25% share for supplier			Exemplary evaluation						
Category	**Criteria**	**Weighting (w)**		**0**	**1**	**2**	**3**	**4**	
Market & supplier related	Market structure	18.62%	There is a large number of suppliers with comparable products			X			There is only one supplier
	Market share	8.67%	Supplier has a large market share (> 40%)		X				Supplier has only a very small market share
	Flagship product	8.67%	Product is main sales driver and market is growing strongly				X		Product contributes very little to supplier's sales, hardly any market growth
Product related	Substitutability	18.11%	There are many comparable products or technologies on the market		X				There are no comparable products, not even in other industries/sectors
	Complexity	9.95%	Very low complexity					X	Very high complexity of the product
	Technology attractiveness	10.46%	Basic technology	X					Pacemaker technology
Cooperation related	Purchase volume	10.20%	High purchase volume by OEM (many vehicle variants, several years)				X		Low purchase volume (few variants, short contract term)
	Risk of a lock-in effect	15.31%	Supplier selection leads to significant lock-in effects and high switching costs					X	No risk of a lock-in effect
$\sum$=		100%							

product-related criteria are substitutability, complexity, and technology attractiveness. According to Simon and Fassnacht (2016), if there are many comparable products or technologies (= high substitutability), this reduces the bargaining power of the supplier and therefore leads to a lower profit share in the proposed model. On the other hand, if the complexity of the product is high, there is a higher effort for programming and troubleshooting. This should lead to a higher profit share from the supplier's perspective. The last product-related factor is technology attractiveness. The factor has proven to be a suitable means of evaluating technologies in the context of innovation cost management. A distinction can be made between pacemaker, key, and basic technologies. Although a pacemaker technology is still at an early stage of development and has very high competitive potential, a basic technology is already integrated in most products. Because here the technology is in the maturity phase, no further major investments are being made. A key technology, on the other hand, is still in the growth phase and can be placed between the other two technology types. Software products are particularly profitable in their maturity phase and have usually already covered the high investments required to create the first copy. Therefore, as the technology becomes less attractive, a smaller share of the profit should flow to the supplier. For a more detailed categorization, see Hartmann (2010). The two remaining cooperation-related influencing variables are the purchase volume and the risk of a lock-in effect. Following Blair and Durrance (2014) a larger purchase volume leads to lower unit prices and thus to a lower profit per unit from the supplier's perspective. Therefore, in the developed model, a higher purchase volume by the OEM is accompanied by a lower profit share for the supplier. If there is a risk of a lock-in effect due to the purchase of the software or if high switching costs would arise in the event of a future change of the supplier, this should lead to a lower profit share accordingly. Once the criteria for the profit split have been identified, a weighting factor for the respective criterion is included in the calculation of the profit split. A pairwise comparison is used to determine the weights, which is not further detailed in this research. Generally, the procedure compares the individual criteria in a square matrix and thus breaks down a complex decision into several small partial decisions. The eight criteria (see Table 2) are rated by five experts from the cost engineering and two experts from the purchasing department at the OEM. Based on this, the supplier's profit share is estimated, and the resulting purchase price is calculated. Figure 2 illustrates the procedure.

Following Figure 2 and Formula 12 a profit split of 10.48% is calculated. The purchase price per voice control is now obtained by multiplying the previously calculated optimal price (p^* = 88.15 €, Formula 11) by the supplier share. As an overall result the "appropriate" purchase price (EK_p) is €9.24.

Criterion i	weighting	Ø Rating (0-4)	Rating in % (Ø / 4)	Rating (%) * weighting (%)
	W_i	$R^{\emptyset}_{i,p}$	$(R^{\emptyset}_{i,p}/R^{max})$	$W_i \times (R^{\emptyset}_{i,p}/R^{max})$
Market structure	18.62%	1.25	31%	0.058195153
Market share	8.67%	1.25	31%	0.027104592
Flagship product	8.67%	1.25	31%	0.027104592
Substitutability	18.11%	1.5	38%	0.067920918
Complexity	9.95%	2.5	63%	0.062181122
Technology attractiveness	10.46%	2	50%	0.052295918
Purchase volume	10.20%	2.25	56%	0.057397959
Risk of a lock-in effect	15.31%	1.75	44%	0.066964286
	100%	1.677		0.419
	Σ	weighted Σ		Σ= ¼ * 1.677

	Customer value (p*)	Purchase price(EK_p)
Speech	88.15€ →	9.24€
Max. Supplier share	0.25	
Supplier share	10.48%	

* 25 % (PS^{max})

FIGURE 2 Profit share and purchase price estimation

$$EK_p = PS_p \cdot p^* = 0.1048 \cdot 88.15€ = 9.24€ \quad \forall p \in P$$

$$PS_p = \frac{1}{R^{max}} \cdot \sum_{i \in I} \left(W_i \cdot R^{\emptyset}_{i,p} \right) \cdot PS^{max} = \frac{1}{4} \cdot 1.677 \cdot 0.25 = 0.419 \cdot 0.25 = 0.1048 \quad \forall p \in P \qquad (12)$$

EK_p purchase price for Product p
PS_p profit split with supplier for Product p (here 0.1048)
p^* optimal end customer selling price (here 88.15 €)
R^{max} maximum achievable rating (here 4)
W_i weighting of criterion i in %.
$R^{\emptyset}_{i,p}$ average expert rating of criterion i for Product p
PS^{max} maximum achievable profit share of supplier (here 0.25)

It has been proved that the value-based model for making plausible purchase prices of software licenses does provide useful results.

4 | LIMITATIONS AND MANAGEMENT IMPLICATIONS

The presented value-oriented evaluation model for licenses is limited to software that can be directly experienced by the end consumer. However, this can be difficult due to the special features of digital goods. Due to the immaterial character of software, it is, in contrast to physical products, only perceptible to a limited extent. The functions of digital products can only be grasped and evaluated through two senses (seeing, hearing). For this reason, digital products are also called trust goods or experience goods (Buxmann et al., 2013). This means that the value of software from the user's point of view is difficult to assess before using it. Purchase decisions for digital goods are therefore always made partly under uncertainty (Atasoy & Morewedge, 2018). This situation makes it difficult to compare software products and complicates the determination of potential customers' willingness to pay. There is also uncertainty about willingness to pay for digital goods for which free alternatives are available. In the case of software with a high degree of innovation, better results may be achieved if respondents have the opportunity to try out the software and learn about its functions.

5 | CONCLUSION

Automobile manufacturers are confronted with rising costs in software purchasing due to an increasing digitalization in the vehicle. However, there is no satisfactory concept for the evaluation of licenses for automotive software in literature and practice. This is surprising against the background of a dynamically increasing share of software in the vehicle and its growing importance as a competitive factor. In order to evaluate offers of software licenses with regard to their price adequacy, new approaches and innovative methods are required. However, it has to be noted that software licenses are very difficult to penetrate in terms of cost analysis. Standard software is characterized by high initial development costs (first copy costs) and marginal duplication costs close to zero. In addition, suppliers of standardized digital goods usually do not determine their prices on a cost basis. Furthermore, the information available from the OEM's point of view is sparse and there is a high heterogeneity of software types used in vehicles. The value-based model presented in this paper is based on the monetary determination of utility values from the user's perspective. This is done with the help of a conjoint analysis. Subsequently, the determined value is converted into a profit-maximizing license price. The validation of the model is illustrated by the example of a speech control system. Based on the results obtained, it can be assumed that the value-based model is suitable for plausibility checks of purchase prices in practice.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

ORCID

Frank Bodendorf https://orcid.org/0000-0002-4784-5516

REFERENCES

Abdat, N., Spruit, M., & Bos, M. (2011). Software as a service and the pricing strategy for vendors. In *Digital product management, technology and practice: Interdisciplinary perspectives* (pp. 154–192). Pennsylvania: IGI Global.

Arbuckle, J. L. (2011). *IBM SPSS Amos 20 user's guide*. Pennsylvania: Amos Development Corporation, SPSS Inc.

Atasoy, O., & Morewedge, C. K. (2018). Digital goods are valued less than physical goods. *Journal of Consumer Research*, *44*(6), 1343–1357. https://doi.org/10.1093/jcr/ucx102

Bala, R., & Carr, S. (2010). Usage-based pricing of software services under competition. *Journal of Revenue and Pricing Management*, *9*(3), 204–216. https://doi.org/10.1057/rpm.2010.12

Balasubramanian, S., Bhattacharya, S., & Krishnan, V. V. (2015). Pricing information goods: A strategic analysis of the selling and pay-per-use mechanisms. *Marketing Science*, *34*(2), 218–234. https://doi.org/10.1287/mksc.2014.0894

Baumeister, A., & Ilg, M. (2004). Entwicklungsbegleitende Kalkulation von Softwareprojekten nach dem Unified Process. *Wirtschaftsinformatik*, *46*(3), 188–195. https://doi.org/10.1007/BF03250936

Baur, A. W., Genova, A. C., Bühler, J., & Bick, M. (2014). Customer is king? A framework to shift from cost-to value-based pricing in software as a service: The case of business intelligence software. In *Conference on e-business, e-services and e-society* (pp. 1–13). Berlin, Heidelberg: Springer.

Bertozzi, M., Broggi, A., & Fascioli, A. (2000). Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous Systems*, *32*(1), 1–16. https://doi.org/10.1016/S0921-8890(99)00125-6

Bilgaiyan, S., Sagnika, S., Mishra, S., & Das, M. (2017). A systematic review on software cost estimation in agile software development. *Journal of Engineering Science & Technology Review*, *10*(4), 51–64.

Blair, R. D., & Durrance, C. P. (2014). Group purchasing organizations, monopsony, and antitrust policy. *Managerial and Decision Economics*, *35*(7), 433–443. https://doi.org/10.1002/mde.2633

Bontis, N., & Chung, H. (2000). The evolution of software pricing: From box licenses to application service provider models. *Internet Research*, *10*(3), 246–255. https://doi.org/10.1108/10662240010331993

Brassel, S., & Gadatsch, A. (2017). Softwarenutzung im Umbruch: Von der Software-Lizenz zum Cloudbasierten Business Process Outsourcing. *HMD Praxis der Wirtschaftsinformatik*, *54*(1), 156–164. https://doi.org/10.1365/s40702-016-0279-9

Brocke, J. V., Simons, A., Niehaves, B., Niehaves, B., Reimer, K., Plattfaut, R., & Cleven, A. (2009). Reconstructing the giant: On the importance of rigour in documenting the literature search process.

Buxmann, P., Diefenbach, H., & Hess, T. (2013). *The software industry: Economic principles, strategies, perspectives*. Berlin, Heidelberg: Springer Science & Business Media. https://doi.org/10.1007/978-3-642-31510-7

Carroll, J. D., & Green, P. E. (1995). Psychometric methods in marketing research: Part I, conjoint analysis. *Journal of Marketing Research*, *32*(4), 385–391.

Clement, R., & Schreiber, D. (2016). *Internet-Ökonomie*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Cousins, P. D., Lawson, B., Squire, B., & Squire, B. (2006). An empirical taxonomy of purchasing functions. *International Journal of Operations & Production Management*, *26*(7), 775–794. https://doi.org/10.1108/01443570610672239

Croux, C., & Dehon, C. (2010). Influence functions of the spearman and Kendall correlation measures. *Statistical Methods & Applications*, *19*(4), 497–515. https://doi.org/10.1007/s10260-010-0142-z

Cusumano, M. A. (2007). The changing labyrinth of software pricing. *Communications of the ACM*, *50*(7), 19–22. https://doi.org/10.1145/1272516.1272531

Darmon, R., & Rouziès, D. (1994). Reliability and internal validity of conjoint estimated utility functions under error-free versus error-full conditions. *International Journal of Research in Marketing*, *11*(5), 465–476. https://doi.org/10.1016/0167-8116(94)00005-0

Dewi, R. S., & Subriadi, A. P. (2017). A modification complexity factor in function points method for software cost estimation towards public service application. *Procedia Computer Science*, *124*, 415–422. https://doi.org/10.1016/j.procs.2017.12.172

Dragicevic, S., Celar, S., & Turic, M. (2017). Bayesian network model for task effort estimation in agile software development. *Journal of Systems and Software*, *127*, 109–119. https://doi.org/10.1016/j.jss.2017.01.027

Elrod, T., Louviere, J. J., & Davey, K. S. (1992). An empirical comparison of ratings-based and choice-based conjoint models. *Journal of Marketing Research*, *29*(3), 368–377. https://doi.org/10.1177/002224379202900307

Foster, E. (2014). *Software engineering: A methodical approach*. New York City: Apress. https://doi.org/10.1007/978-1-4842-0847-2

Frohmann, F. (2018). *Digitales pricing*. Wiesbaden: Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-22573-5

Greco, S., Matarazzo, B., & Słowiński, R. (2004). Axiomatic characterization of a general utility function and its particular cases in terms of conjoint measurement and rough-set decision rules. *European Journal of Operational Research*, *158*(2), 271–292. https://doi.org/10.1016/j.ejor.2003.06.004

Green, P. E., & Krieger, A. M. (1996). Individualized hybrid models for conjoint analysis. *Management Science*, *42*(6), 850–867. https://doi.org/10.1287/mnsc.42.6.850

Green, P. E., Krieger, A. M., & Agarwal, M. K. (1991). Adaptive conjoint analysis: Some caveats and suggestions. *Journal of Marketing Research*, *28*(2), 215–222. https://doi.org/10.1177/002224379102800208

Gustafsson, A., Herrmann, A., & Huber, F. (Eds.) (2007). *Conjoint measurement: Methods and applications*. Berlin, Heidelberg: Springer Science & Business Media. https://doi.org/10.1007/978-3-540-71404-0

Handfield, R., Walton, S. V., Sroufe, R., & Melnyk, S. A. (2002). Applying environmental criteria to supplier assessment: A study in the application of the analytical hierarchy process. *European Journal of Operational Research*, *141*(1), 70–87. https://doi.org/10.1016/S0377-2217(01)00261-2

Hansen, E., & Bush, R. J. (1999). Understanding customer quality requirements: Model and application. *Industrial Marketing Management*, *28*(2), 119–130. https://doi.org/10.1016/S0019-8501(98)00007-8

Harmon, R., Demirkan, H., Hefley, B., & Auseklis, N. (2009). Pricing strategies for information technology services: A value-based approach. In 2009 42nd Hawaii International Conference on System Sciences (HICSS) (pp. 1-10). IEEE.

Hartmann, M. (2010). *Technology cost analysis*. In Innovation performance accounting. (pp. 271–283). Berlin, Heidelberg: Springer.

Hoe, S., & Diltz, J. D. (2012). A real options approach to valuing and negotiating licensing agreements. *The Quarterly Review of Economics and Finance*, *52*(3), 322–332. https://doi.org/10.1016/j.qref.2012.04.005

Hoisl, B. (2018). *Produkte digital-first denken: Wie Unternehmen softwarebasierte Produktinnovation erfolgreich gestalten*. Wiesbaden: Springer-Verlag. https://doi.org/10.1007/978-3-658-23051-7

Hylving, L. (2015). Competing values in the era of digitalization. In HICSS '15: Proceedings of the 2015 48th Hawaii International Conference on System Sciences (pp. 4161–4170). IEEE. https://doi.org/10.1109/HICSS.2015.499

Irgens, C., Tawfik, S., & Landryova, L. (2004). Evaluating a software costing method based on software features and case based reasoning. In *International Conference on Information Technology for Balanced Automation Systems* (pp. 467–474). Boston, MA: Springer.

Jia, K., Liao, X., & Feng, J. (2018). Selling or leasing? Dynamic pricing of software with upgrades. *European Journal of Operational Research*, *266*(3), 1044–1061. https://doi.org/10.1016/j.ejor.2017.10.063

Kalish, S., & Nelson, P. (1991). A comparison of ranking, rating and reservation price measurement in conjoint analysis. *Marketing Letters*, *2*(4), 327–335. https://doi.org/10.1007/BF00664219

Kashyap, D., & Misra, A. K. (2013). Software development cost estimation using similarity difference between software attributes. In ISDOC '13: Proceedings of the 2013 International Conference on Information Systems and Design of Communication (pp. 1–6). https://doi.org/10.1145/2503859.2503860

Kephart, J. O., Hanson, J. E., & Greenwald, A. R. (2000). Dynamic pricing by software agents. *Computer Networks*, *32*(6), 731–752. https://doi.org/10.1016/S1389-1286(00)00026-8

Keshta, I. M. (2017). Software cost estimation approaches: A survey. *Journal of Software Engineering and Applications*, *10*(10), 824–842. https://doi.org/10.4236/jsea.2017.1010046

Kittlaus, H. B., & Clough, P. N. (2009). *Software product management and pricing: Key success factors for software organizations*. Berlin, Heidelberg: Springer Science & Business Media. https://doi.org/10.1007/978-3-540-76987-3

Koppius, O. (1999). Dimensions of intangible goods. In Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences. 1999. HICSS-32. Abstracts and CD-ROM of Full Papers (pp. 7-pp). IEEE. https://doi.org/10.1109/HICSS.1999.772916

Köthner, R. (2012). Immaterielle Vermögenswerte: Praxis der Bewertung im Rahmen einer Kaufpreisallokation nach IFRS. *Controlling & Management*, *56*(1), 57–65. https://doi.org/10.1365/s12176-012-0146-7

Krcmar, H. (2015). Informations management. In *Informations management* (pp. 85–111). Berlin, Heidelberg: Springer Gabler.

Kübler, R., Pauwels, K., Yildirim, G., & Fandrich, T. (2018). App popularity: Where in the world are consumers most sensitive to price and user ratings? *Journal of Marketing*, *82*(5), 20–44. https://doi.org/10.1509/jm.16.0140

Lehmann, S., & Buxmann, P. (2009). Pricing strategies of software vendors. *Business & Information Systems Engineering*, *1*(6), 452–462. https://doi.org/10.1007/s12599-009-0075-y

Lehmann, S., Draisbach, T., Buxmann, P., & Dörsam, P. (2012). Pricing of software as a service—An empirical study in view of the economics of information theory. In *International Conference of Software Business* (pp. 1–14). Berlin, Heidelberg: Springer.

Liu, Y., Cheng, H. K., Tang, Q. C., & Eryarsoy, E. (2011). Optimal software pricing in the presence of piracy and word-of-mouth effect. *Decision Support Systems*, *51*(1), 99–107. https://doi.org/10.1016/j.dss.2010.11.032

Llopis-Albert, C., Rubio, F., & Valero, F. (2020). Impact of digital transformation on the automotive industry. *Technological Forecasting and Social Change*, *162*, 120343.

Miller, J., Wienke, S., Schlottke-Lakemper, M., Meinke, M., & Müller, M. S. (2018). Applicability of the software cost model COCOMO II to HPC projects. *International Journal of Computational Science and Engineering*, *17*(3), 283–296. https://doi.org/10.1504/IJCSE.2018.095849

Mueller, M., Hoermann, K., Dittmann, L., & Zimmer, J. (2008). *Automotive SPICE in practice: Surviving implementation and assessment*. San Rafael: Rocky Nook, Inc. ISBN-13 : 978-1933952291.

Murtojarvi, M., Jarvinen, J., Johnsson, M., Leipala, T., & Nevalainen, O. S. (2007). Determining the proper number and price of software licenses. *IEEE Transactions on Software Engineering*, *33*(5), 305–315. https://doi.org/10.1109/TSE.2007.1003

Natter, M., & Feurstein, M. (2002). Real world performance of choice-based conjoint models. *European Journal of Operational Research*, *137*(2), 448–458. https://doi.org/10.1016/S0377-2217(01)00147-3

Nestler, A. (2008). Ermittlung von Lizenzentgelten. https://www.valnes.de/_Resources/Persistent/578b701305dd307ed034885fe8c9eb0a00cca563/Aufsatz_DAN_Lizenzen_BB_37_08.pdf

Nestler, A. (2013). Angemessene Lizenzbewertung anhand der Profit-Split-Methode: Was man von der Knoppe-Formel lernen kann. *Bilanzrecht Und Betriebswirtschaft*, (34), 2027–2029.

Nestler, A. (2017). Die finanzorientierte Bewertung von Software. M&A REVIEW, 01–02.

Ojala, A. (2016). Adjusting software revenue and pricing strategies in the era of cloud computing. *Journal of Systems and Software*, *122*, 40–51. https://doi.org/10.1016/j.jss.2016.08.070

Pospieszny, P., Czarnacka-Chrobot, B., & Kobylinski, A. (2018). An effective approach for software project effort and duration estimation with machine learning algorithms. *Journal of Systems and Software*, *137*, 184–196. https://doi.org/10.1016/j.jss.2017.11.066

Pullman, M. E., Dodson, K. J., & Moore, W. L. (1999). A comparison of conjoint methods when there are many attributes. *Marketing Letters*, *10*(2), 125–138. https://doi.org/10.1023/A:1008036829555

Qualtrics. (2018). Provo, Utah, USA: Qualtrics. Retrieved from https://www.qualtrics.com

Rohitratana, J., & Altmann, J. (2012). Impact of pricing schemes on a market for software-as-a-service and perpetual software. *Future Generation Computer Systems*, *28*(8), 1328–1339. https://doi.org/10.1016/j.future.2012.03.019

Roy, R. (2003). Cost engineering: why, what and how?. https://dspace.lib.cranfield.ac.uk/bitstream/handle/1826/64/cost%20engineering%20why%20what%20and%20how.pdf?sequence=1&isAllowed=y

Sainio, L. M., & Marjakoski, E. (2009). The logic of revenue logic? Strategic and operational levels of pricing in the context of software business. *Technovation*, *29*(5), 368–378. https://doi.org/10.1016/j.technovation.2008.10.009

Schawel, C., & Billing, F. (2018). BCG-matrix. In *Top 100 management tools* (pp. 41–43). Wiesbaden: Springer Gabler.

Schneider, S. (2012). The dirty little secret of software pricing. RTI Whitepaper, available at: www.rti.com/whitepapers/Dirty_Little_Secret.pdf (accessed 25 February 2021).

Scholl, A., Manthey, L., Helm, R., & Steiner, M. (2005). Solving multiattribute design problems with analytic hierarchy process and conjoint analysis: An empirical comparison. *European Journal of Operational Research*, *164*(3), 760–777. https://doi.org/10.1016/j.ejor.2004.01.026

Schwarzer, E. (2016). Die strategische Betrachtung eines neuen Preis-und Lizenzmodells in der Softwarebranche. In *Informationstechnologien als Wegbereiter für den steuerberatenden Berufsstand* (pp. 187–205). Berlin, Heidelberg: Springer Gabler.

Sequeira, S., & Lopes, E. (2015). Simple method proposal for cost estimation from work breakdown structure. *Procedia Computer Science*, *64*, 537–544. https://doi.org/10.1016/j.procs.2015.08.559

Simon, H., & Fassnacht, M. (2016). *Preismanagement: Strategie-Analyse-Entscheidung-Umsetzung*. Wiesbaden: Springer-Verlag.

Singh, T., Singh, R., & Mishra, K. K. (2018). Software cost estimation using environmental adaptation method. *Procedia Computer Science*, *143*, 325–332. https://doi.org/10.1016/j.procs.2018.10.403

Specht, G. (2013). *Grundlagen der Preisführerschaft: eine betriebswirtschaftliche Betrachtung unter besonderer Berücksichtigung des Einzelhandels*. Wiesbaden: Springer-Verlag. https://doi.org/10.1007/978-3-322-91036-3

Steinle, C., Schiele, H., & Ernst, T. (2014). Information asymmetries as antecedents of opportunism in buyer-supplier relationships: Testing principal-agent theory. *Journal of Business-to-Business Marketing*, *21*(2), 123–140. https://doi.org/10.1080/1051712X.2014.903457

Symons, C. R. (1988). Function point analysis: Difficulties and improvements. *IEEE Transactions on Software Engineering*, *14*(1), 2–11. https://doi.org/10.1109/32.4618

Tam, C., da Costa Moura, E. J., Oliveira, T., & Varajão, J. (2020). The factors influencing the success of on-going agile software development projects. *International Journal of Project Management*, *38*(3), 165–176. https://doi.org/10.1016/j.ijproman.2020.02.001

Töytäri, P., Rajala, R., & Alejandro, T. B. (2015). Organizational and institutional barriers to value-based pricing in industrial relationships. *Industrial Marketing Management*, *47*, 53–64. https://doi.org/10.1016/j.indmarman.2015.02.005

Tripathi, A., Mishra, K. K., Tiwari, S., & Kumar, N. (2018). Improved software cost estimation models: A new perspective based on evolution in dynamic environment. *Journal of Intelligent & Fuzzy Systems*, *35*(2), 1707–1720. https://doi.org/10.3233/JIFS-169707

Tsunoda, M., Monden, A., Matsumoto, K., Ohiwa, S., & Oshino, T. (2012, October). Analysis of attributes relating to custom software price. In 2012 Fourth International Workshop on Empirical Software Engineering in Practice (pp. 16-22). IEEE. https://doi.org/10.1109/IWESEP.2012.19

Venkataiah, V., Mohanty, R., & Nagaratna, M. (2019). Prediction of software cost estimation using spiking neural networks. In *Smart intelligent computing and applications* (pp. 101–112). Singapore: Springer.

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly*, *26*(2), xiii–xxiii. https://doi.org/10.2307/4132319

Wieczorek, I. (2003). Verbesserte Software Kostenschätzung-Eine robuste und interpretierbare Modellierungsmethode und deren umfassende empirische Überprüfung. *Ausgezeichnete Informatikdissertationen*, *2001*. https://subs.emis.de/LNI/Dissertation/Dissertation2/GI-Dissertations.02-18.pdf

Wirtz, H. (2010). Die monetäre Bewertung technologiebasierter immaterieller Vermögenswerte. *Controlling & Management*, *54*(4), 224–228. https://doi.org/10.1007/s12176-010-0074-3

Wu, S., Wortmann, H., & Tan, C. W. (2014, August). A pricing framework for software-as-a-service. In fourth edition of the International Conference on the Innovative Computing Technology (INTECH 2014) (pp. 152-157). IEEE. https://doi.org/10.1109/INTECH.2014.6927738

Zhang, Z. (2020). Competitive pricing strategies for software and SaaS products. *Information & Management*, *57*(8), 103367. https://doi.org/10.1016/j.im.2020.103367

How to cite this article: Bodendorf, F., Lutz, M., & Franke, J. (2021). Valuation and pricing of software licenses to support supplier–buyer negotiations: A case study in the automotive industry. *Managerial and Decision Economics*, *42*(7), 1686–1702. https://doi.org/10.1002/mde.3336

APPENDIX: OVERVIEW OF PUBLICATIONS

Publication			Concept										
						Valuation of intangible products				Estimation of development costs from the supplier's perspective			
										Algorithmic estimation models			
No.	Author/year	Title	Pricing strategies in the software industry	Mathematical modeling/ simulation	Evaluation of software prices from a customer perspective	Market price-oriented	Capital value-oriented	Cost-oriented	Profit split	COCOMO	Function point	Prediction model	Analogy-oriented/ similarity-based
1	Abdat et al. (2011)	Software as a service and the pricing strategy for vendors	x										
2	Bala and Carr (2010)	Usage-based pricing of software services under competition		x									
3	Balasubramanian et al. (2015)	Pricing information goods: A strategic analysis of the selling and pay-per-use mechanisms		x									
4	Baumeister and Ilg (2004)	Entwicklungsbegleitende Kalkulation von Softwareprojekten nach dem unified process											x
5	Baur et al. (2014)	Customer is king? A framework to shift from cost- to value-based pricing in software as a service	x										
6	Bilgaiyan et al. (2017)	A systematic review on software cost estimation in agile software development	x	x	x	x	x	x		x	x	x	x
7	Bontis and Chung (2000)	The evolution of software pricing: From box licenses to application service provider models	x										
8	Brassel and Gadatsch (2017)	Softwarenutzung im Umbruch: Von der Software-Lizenz zum Cloudbasierten business process outsourcing	x										
9	Buxmann et al. (2013)	Die Softwareindustrie	x										

Publication			Concept										
						Valuation of intangible products				Estimation of development costs from the supplier's perspective			
										Algorithmic estimation models			
No.	Author/year	Title	Pricing strategies in the software industry	Mathematical modeling/ simulation	Evaluation of software prices from a customer perspective	Market price-oriented	Capital value-oriented	Cost-oriented	Profit split	COCOMO	Function point	Prediction model	Analogy-oriented/ similarity-based
10	Clement and Schreiber (2016)	Internet-Ökonomie: Grundlagen und Fallbeispiele der vernetzten Wirtschaft.	x										
11	Cusumano (2007)	The changing labyrinth of software pricing	x										
12	Dewi and Subriadi (2017)	A modification complexity factor in function points method for software cost estimation towards public service application		x				x		x	x		
13	Dragicevic et al. (2017)	Bayesian network model for task effort estimation in agile software development		x				x				x	
14	Foster (2014)	Software economics	x							x	x		
15	Frohmann (2018)	DigitalesPricing	x										
16	Harmon et al. (2009)	Pricing strategies for information technology services: A value-based approach	x										
17	Hoisl (2018)	Produkte digital-first denken	x										
18	Irgens et al. (2004)	Evaluating a software costing method based on software features and case based reasoning											x
19	Jia et al. (2018)	Selling or leasing? Dynamic pricing of software with upgrades	x	x	x	x	x	x					
20	Kashyap and Misra (2013)	Software cost estimation using similarity difference between software attributes											x
21	Kephart et al. (2000)	Dynamic pricing by software agents		x									

(Continues)

Publication			Concept										
						Valuation of intangible products				Estimation of development costs from the supplier's perspective			
										Algorithmic estimation models			
No.	Author/year	Title	Pricing strategies in the software industry	Mathematical modeling/ simulation	Evaluation of software prices from a customer perspective	Market price-oriented	Capital value-oriented	Cost-oriented	Profit split	COCOMO	Function point	Prediction model	Analogy-oriented/ similarity-based
22	Kittlaus and Clough (2009)	Software product management and pricing	x										
23	Köthner (2012)	Immaterielle Vermögenswerte: Praxis der Bewertung im Rahmen einer Kaufpreisallokation nach IFRS				x	x	x					
24	Krcmar (2015)	Informationsmanagement	x										
25	Keshta (2017)	Software cost estimation approaches: a survey	x	x	x	x	x	x	x	x	x	x	x
26	Lehmann et al. (2012)	Pricing of software as a service—An empirical study in view of the economics of information theory	x										
27	Lehmann and Buxmann (2009)	Preisstrategien von Softwareanbietern	x										
28	Liu et al. (2011)	Optimal software pricing in the presence of piracy and word-of-mouth effect	x	x									
29	Miller et al. (2018)	Applicability of the software cost model COCOMO II to HPC projects		x				x		x	x		
30	Murtojarvi et al. (2007)	Determining the proper number and price of software licenses		x									
31	Nestler (2008)	Ermittlung von Lizenzentgelten				x	x	x	x				
32	Nestler (2013)	Angemessene Lizenzbewertung anhand der Profit-Split-Methode							x				
33	Nestler (2017)	Die finanzorientierte Bewertung von Software				x	x	x		x	x		
34	Ojala (2016)	Adjusting software revenue and pricing strategies in the era of cloud computing	x										

Publication			Concept										
						Valuation of intangible products				Estimation of development costs from the supplier's perspective			
										Algorithmic estimation models			
No.	Author/year	Title	Pricing strategies in the software industry	Mathematical modeling/ simulation	Evaluation of software prices from a customer perspective	Market price-oriented	Capital value-oriented	Cost-oriented	Profit split	COCOMO	Function point	Prediction model	Analogy-oriented/ similarity-based
35	Pospieszny et al. (2018)	An effective approach for software project effort and duration estimation with machine learning algorithms		x								x	
36	Rohitratana and Altmann (2012)	Impact of pricing schemes on a market for software-as-a-service and perpetual software	x	x		x	x	x					
37	Roy (2003)	Cost engineering why what and how?								x		x	x
38	Sainio and Marjakoski (2009)	The logic of revenue logic? Strategic and operational levels of pricing in the context of software business	x										
39	Schneider (2012)	The dirty little secret of software pricing	x										
40	Schwarzer (2016)	Die strategische Betrachtung eines neuen Preis- und Lizenzmodells in der Softwarebranche	x										
41	Sequeira and Lopes (2015)	Simple method proposal for cost estimation from work breakdown structure											x
42	Singh et al. (2018)	Software cost estimation using environmental adaptation method								x		x	
43	Tripathi et al. (2018)	Improved software cost estimation models								x		x	
44	Tsunoda et al. (2012)	Analysis of attributes relating to custom software price			x								
45	Venkataiah et al. (2019)	Prediction of software cost estimation using spiking neural networks		x				x				x	

(Continues)

Publication			Concept										
						Valuation of intangible products				Estimation of development costs from the supplier's perspective			
										Algorithmic estimation models			
No.	Author/year	Title	Pricing strategies in the software industry	Mathematical modeling/ simulation	Evaluation of software prices from a customer perspective	Market price-oriented	Capital value-oriented	Cost-oriented	Profit split	COCOMO	Function point	Prediction model	Analogy-oriented/ similarity-based
46	Wieczorek (2003)	Verbesserte Software Kostenschätzung										x	
47	Wirtz (2010)	Die monetäre Bewertung technologiebasierter immaterieller Vermögenswerte				x	x	x					
48	Wu et al. (2014)	A pricing framework for software-as-a-service	x										
49	Zhang (2020)	Competitive pricing strategies for software and SaaS products	x	x	x		x						