Towards a Pricing Method for Software Product Lines

First Steps, Problems, Needs, and Research Agenda

Johannes Müller

Information Systems Institute, University of Leipzig

1 Introduction

Software product lines (spls) are an emerging paradigm, which can help to increase the efficiency of software development, thereby, decrease the development costs for a single system within a given domain. That is achieved by exploiting similarities between systems of a whole family of systems while managing their variability in a systematic manner. The orientation towards spls is a *strategical* decision, which affects the whole organization. This is because the setup of an spl needs initially an high investment. Afterwards, the effort to build a system out of the spl is significantly smaller compared to a single system development (cf. fig. 1). Hence, a number of systems needs to be build – and of course sold – out of the spl until the break even point, the point where an investment generates positive profit, is reached (cf. fig. 1).

The profit (Π) is defined as the difference between revenue (R) and cost (C)

$$\Pi = R - C.$$

This definition implies two possibilities to expand profit: The first one is to reduce costs, which is a major concern of spl research (Cohen 1999; Czarnecki 2005; Krueger 2006; v. d. Linden et al. 2007; Schmid and Verlage 2002; Schmid 2002a; Whitey 1996). The other lever is to raise the revenue, which is made by increasing the sales or by implementing a higher price on the market without disproportionally reducing the sales. This approach seems to be of minor importance for the spl research community even though the price is a primal mean to recover the invested effort. Sewerjuk (2008) directly addresses the pricing of spls, however, without proposing a comprehensive method for price setting. Others (Chastek 2009; Helferich et al. 2006; Helferich et al. 2006a; Kang et al. 2002; v. d. Linden et al. 2007; Müller 2009) discuss some marketing concerns of spls but without directly bespeaking pricing.

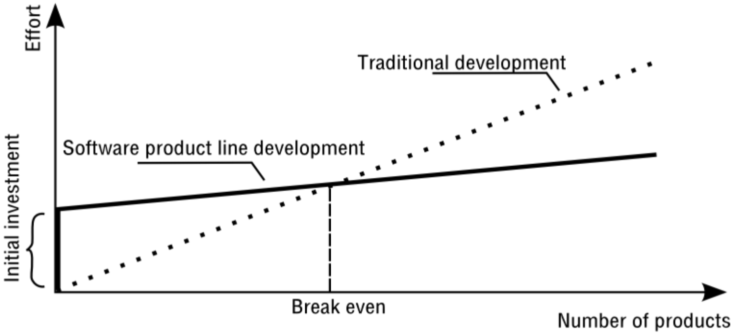


Figure 1: Break even point for spl investment (Schmid and Verlage 2002, p. 51)

However, what is the profit-optimal price for a spl vendor? To answer this, one is faced with the special properties of software or digital goods in general (Buxmann et al. 2008, p. 1). Amongst others there are two important facts: At first software raises high fixed costs for the first copy but low marginal costs for any further copy sold. The fixed costs are due to analysis, design, implementation, and testing of the software system. The variable costs for any further copy mostly raised by pressing a software CD or DVD, printing the manual or maybe running a server, which offers the software for download (Shy 2001, p. 53). The fixed costs are mostly *sunken costs* – that means the costs cannot be influenced after they are accrued (Buxmann et al. 2008, p. 19).

The constellation of high fixed cost and low marginal cost implies that cost based pricing is not feasible. Because every price above zero results in, caused by marginal costs near zero, a positive total contribution margin (Shapiro and Varian 1998, p. 24). Another pricing approach is required. One approach is to set the price according to the customers' idea of a desired price level. Albeit the sales quantity can be increased, the overall profit would be forced down because the price would generally be under the true willingness to pay of the customers. To avoid this hassles value based pricing has been proposed (Kotler and Armstrong 2006; Nagle and Hogan 2006; Shapiro and Varian 1998).

Value based pricing means that the price of a product is set according to the *true* value it has for the customers. Optimally each customer pays exactly the price he or she is willing to pay – the prices are totally differentiated. However, the question arise how to implemented value base pricing in practice? In order to achieve this a method needs to determine:

- the customer-attached value of the product,
- the market segments to address,
- the structure of the price,
- the appropriate price setting procedures and tactics and,
- has to integrate current state-of-the-art pricing methods and tools.

The last point addresses the extensive body of literature that comprises several aspects of pricing (e.g. Belleflamme 2006; Deneckere and McAfee 1995; Kuenne 1998; McAfee 2007; Monroe 1990; Phlips 1983; Vives 2000; Wilson 1993). However, each addresses the competitive forces (i.e. customers, competitors, suppliers, entrants,substitutes) (Porter 1998) differently and hence cannot be mixed without considering existing interdependencies. The orientation towards spls is a *strategic* decision. This has to be complemented by a *strategic* pricing approach, which raises the profitability of a spl over its whole life cycle. A generic approach, which firstly, addresses value based pricing strategically and, secondly, incorporates different pricing tactics in a comprehensive framework is suggested by Nagle and Hogan (2006). The framework lacks of a formal description and a specialization for spls. Hence, it needs a formalization and a specialization to spls.

To get a pricing method which is applicable for as much as possible spl practices a general approach should serve as a basis for the description of the dependencies between pricing and spl practices. Clements and Northrop (2007) describe such general practices.

The depicted problem area implies some challenges which have to be mastered. First of all, the pricing method of Nagle and Hogan (2006) needs a formal description in order to relate its activities to roles and processes of spl practices. One promising way to do so is method engineering (Brinkkemper 1996). Second of all, the specialized pricing method needs to be integrated into the overall spl development process.

The paper is going to present the key points of a formal description of the framework, specialize it, and relate pricing with existing spl practices. A conclusion summarizes the discussion and points to ongoing research activities.

2 The Pricing Framework of Nagle and Hogan

Nagle and Hogan (2006) propose a framework to implement value based pricing. The application of such a strategy requires:

- data about competitor's offers and its prices,
- data about the customers and their product's valuation as well as
- an understanding on how the value differs across market segments.

This enables setting unique prices for each market segment. The strategy is proactive because it educates the customers and also the competitors instead of reacting to their moves. Due to existing information asymmetries between customers and producers, customers are often not aware of the true value a software product spends. Hence, it is crucial to communicate its value to them. Many customers are well informed about price rules and margins and pursue aggressive negotiation tactics. Thus, consistent and clear pricing and discounting policies are pivotal.

A value-based pricing strategy consists of five levels of activities, which are based on each other (Nagle and Hogan 2006, p. 16). On top is the final price level. However, the determination of this level entails four more levels which lay the basis.

The floor level is laid by value creation. A firm has to gain a deep understanding on how its products are differentiated to the offers of its competitors and how they create value for its customers. Additionally, it is decisive to know how the product creates value across different market segments.

If sufficient knowledge about a product's value has been built and market segments and their properties are identified, the next step is to align the price with the value delivered to the different segments. Therefore, two techniques have been described, namely price *metrics* and *fences*. Price metrics are the units by which a price is applied to a product. Fences are used to separate different segments from each other. They are criteria that customers must meet to qualify for discounts and reflect the value as well as the cost created by the sold product. For pricing of spls mainly bundling and tie-ins are interesting, but segmentation by customer characteristics or geographical regions are also promising tactics.

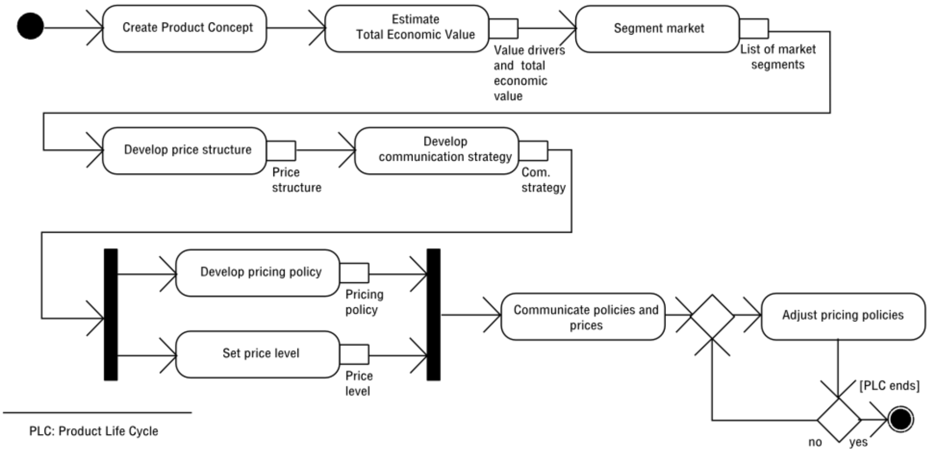


Figure 2: Process to implement a value-based pricing strategy

It is not sufficient just to implement metrics and fences which reflect the value and the segments of the considered product. In particular the prices must be justified based on the value the product creates for different customers. If the value communication fails, price sensitivity will increase. This holds not only for private customers but also for business customers. Business purchases are largely driven by economic value the product spends. Hence, the sales force needs effective tools to communicate the true value of the product. Such tools range from selling sheets to product customization tools (Nagle and Hogan 2006, p. 21). Domain specific languages help to realize a product customization tools for customers as well as for the sales force.

Pricing involves the management of the expectations of customers. These expectations are affected by the company's commitment to enforce pricing policies. Pricing policies are defined by an pricing steering committee, which consists of managers of all departments affected by pricing decisions. If the policies are only weak enforced, customers are taught to aggressively negotiate on prices.

There is one ultimate goal for pricing: align it to the competitive forces in order to maximize profitability. The final price level is influenced by several factors. The selling volume, the cost to cover, the market share, competitors, and the expected long-term profitability. These different factors are differently weighted by the several company departments. For example the marketing department weights long-term profitability and customer satisfaction higher than the finance department which on the other hand weights costs to cover higher. It is vital to incorporate all these sources systematically into one decision model.

A overview about a pricing process for spls derived from the findings Nagle and Hogan (2006) depicts fig. 2. Even though the description of the model references spls, the activities of the model purely focus on price setting. How to incorporate these activities into the spl practices is subject of the next part. Which is going to present the idea of spls at a glance.

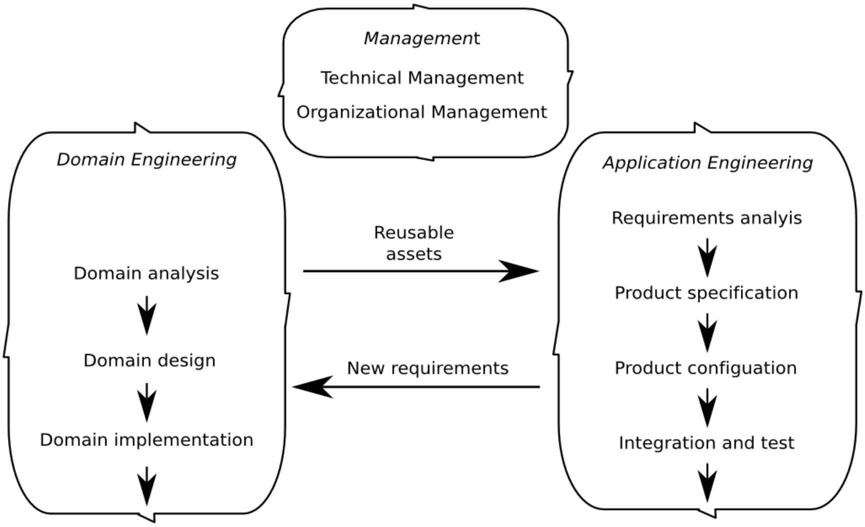
3 Software Product Lines

Spl practices increase the efficiency of software development by systematically exploiting commonalities between software systems of a family of related systems while managing their differences in a systematical manner (Clements and Northrop 2007, pp. 5–6). It is important to distinguish between a *product line* and a *product family*. A *product line* “is a group of products sharing a common, managed set of features that satisfy the specific needs of a selected market” (Whitey 1996, p. 15). On the other hand a *product family* “is a group of products that can be built from a common set of assets” (Whitey 1996, p. 13). Sometimes a product line is called a *marketed software product line* and a product family is called an *engineered software product line* (Helferich et al. 2006a). To foster reuse a product line needs to be based on a product family.

Several methodologies to develop software within a spl exist (Bayer et al. 1999, Böckle et al. 2005; Kang et al. 1990; Kang et al. 1998; Czarnecki and Eisenecker 2000; Schmid 2002). They all comprise three main processes: *domain engineering*, *application engineering*, and *management* (cf. fig. 3) (Clements and Northrop 2007).

The domain engineering – sometimes also referred to as core asset development (Clements and Northrop 2007) or product line engineering (Kang et al., 2003) – is development for reuse. Within this process the domain of interest is analyzed, designed, and finally implemented (Czarnecki 2005). The domain analysis process serves two important purposes. The first is to select and to define – to scope – the domain of focus (Schmid 2002) and the second is to collect relevant information and organize it in a coherent domain model. One part of the domain model describes common and variable features, mostly in form of a feature model. A feature model is used to describe the commonalities and the variability of systems in a domain by means of features – "a distinguishable characteristic of a concept that is relevant to some stakeholder of the concept" (Czarnecki and Eisenecker 2000, p. 38). The result of the domain design is a common architecture for a family of systems and a production plan, which describes how to produce concrete products out of the core assets. Finally, the architecture, the reusable components, and the production plan are implemented in the domain implementation activity.

Application engineering – sometimes also referred to as application development (Clements and Northrop 2007) or product development (Kang et al. 2003) –



Fi

Figure 3: Software product line practices

is development with reuse. Within this process concrete products are produced by means of customers requirements which are used to specify the desired product. Afterwards, this specification can be used to manually, semi-automatically, or automatically configure a system (Cohen 1999). After configuration the system is tested and integrated into the production environment.

The third process is management. Clements and Northrop (2007, pp. 45) identify two main categories: *technical management* and *organizational management*. Technical management oversees domain engineering as well as application engineering and tracks the development progress. Organizational management is responsible for allocating resources and – eventually – for the overall entrepreneurial success of the company. It has to integrate "development, production, and marketing of products that meet customer needs" (Niehaus et al., 2005). Furthermore it is part of the interface to the customer and therefore responsible for analyzing markets, define roles in the specification process, implement processes to govern customer interactions, and finally selling the configured and tested product (Clements and Northrop 2007, pp. 219, 236).

4 Incorporate Pricing into Software Product Line Practices

With the outline of the specialized pricing method and a quick survey on spl practices at hand the combination of both can be approached. Figure 4 depicts the approach. The general pricing method of Nagle and Hogan (2006) is deductively specialized for spls in sec. 3. Afterwards, this specialization is connected to the spl practices. This results in specialized practices for spl development.

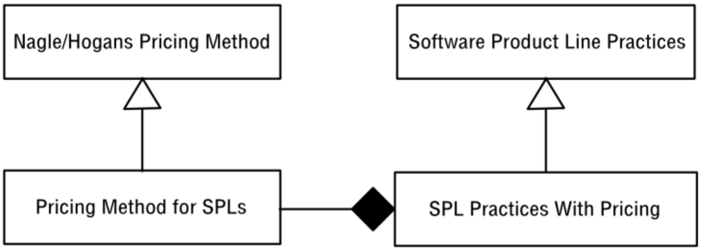

Nagle/Hogans Pricing Method
Software Product Line Practices
Pricing Method for SPLs
SPL Practices With Pricing

Figure 4: Combination of pricing and spl practices depicted as a UML class diagram

The subject of this section is the compositional dependency in fig. 4, which implies that the activities of the pricing method need to be mapped to the subprocesses, activities, and roles of the spl practices. These mappings are described next.

Concept creation, value estimation, and market segmentation are activities that match to the activity domain analysis. While concept creation is part of scoping, the value estimation has to be performed within the domain modeling activity because in this phase all relevant stakeholders are at hand. Additional market research should be done by the organizational management in subprocess management to verify the results of the first steps. With an idea about the different valuation of the spl features, the next step is to segment the market. This is mainly done by the organizational management as part of the management subprocess. Nevertheless, it is worthwhile to consult the domain analysts to get feedback, to double check the results, and finally to gave the domain analysts the newly gathered data to refine the domain model. Hence, the feedback loop has to be implemented between the scoping activity and the domain modeling activity of the domain engineering sub process. Before the features of the spl can be implemented, the price structure needs to be developed. A price structure at hand gives an idea about the possible profit margins of the features to implement and therefore a good rationale to prioritize the implementation of the features. The development of an appropriate communication strategy is in the responsibility of the organizational management and is approached after the development of the price structure and in parallel to the domain implementation activity.

The rules of the pricing policy are defined by the pricing steering committee, which is part of the organizational management. Additional relations depend on the type of product configuration. If the product configuration is semi-automatically or even automatically, pricing has also be automated to preserve the scalability of the whole spl approach. Hence, within domain design it must also be considered to design a tool which organizes the price components of all features. The tool needs to allow price differentiation between market segments by considering feature selection. Based on a given feature selection, a pricing policy, and a identified market segment the tool needs to have the capability to calculate a final price for a concrete product of a spl. This tool is a software artifact and needs to be implemented during domain implementation. The pricing policy as well as the tool are used while product specification to negotiate the price and the received value (i.e. the product features) with the customer in the application engineering sub process.

The final price level for the different features and feature combinations for the identified market segments are based on the data collected before, as activity of the management sub process by the organizational management. To optimize the price organizational management needs feedback from the application engineering sub process about the customers reaction on the prices. If this feedback cycle ought to be automated and an automated price optimization technique (Nagle and Hogan 2006, pp. 140) is desired, this has to be considered during domain design and domain implementation.

The final activity is to communicate the prices and price policies to external as well as to internal stakeholders. This is part of the management sub process and done by the organizational management du to its relations to both groups of stakeholders and its authority to enforce the pricing decisions.

The feedback loop to adjust the pricing policies is triggered by the sales force and consultants, which are both part of organizational management. Consultants are also involved in the application engineering process because of the support they give to customers during product specification.

5 Conclusion and Future Work

The previous section presents a promising combination of pricing and spl practices. Currently, no integrated approach for pricing spls is available. Thus several problems need to be solved until such an approach is present.

The above mentioned mapping relies on the description of Clements and Northrop (2007). As stated before there are several, detailed spl engineering methods described in the literature. It has to be investigated whether and how the described mappings could be integrated into these methods as well.

The specialized pricing method presented in this paper considers only the first phase of a software product's life cycle. It seems to be important to analyze how a pricing strategy covers transformations between all life cycle phases (Niehaus et al. 2005). Especially software maintenance is a emerging source of profit for todays software vendors. Strategies are required, which help to engage customers on the costs of evolving a spl. Initial ideas from the traditional software industry are maintenance contracts. Against a fee the vendor guarantees to maintain the sold software for a period of time.

Particularly spls with many features generate high effort to assemble concrete products out of the core assets. Hence the assembling has to be automated or at least semi-automated to reduce this effort. In such situations the effort can be tremendous to decide for a concrete derived product what price aligns to the pricing strategy and follows the price setting policies. To preserve the scalability reached through the automation of the product assembly it is crucial to automate the management, calculation, and tracking of the prices as well. The question is, where to provide the prices of the features for the different segments and how to design a tool which helps managing the prices in a comprehensible way. A promising way are feature models (Kang et al. 1990).

Finally, the last question is, how to ensure that the described method is feasible and will effectively increase the overall profit of a company? Empirical evidence is needed to convince the industry of the helpfulness of the described approach. Pricing comprises critical and even risky decisions for every business. Therefore it will be unlikely to find a firm that bears the risk of the first adopter to implement a case study. Other ways have to be found to generate confidence. One idea is to set up an experiment based on the findings and approaches of the field of experimental economics.

To summarize, the paper presents a method to price products of spls, which is based on a general pricing method. This method is mapped to processes and activities of spl practices. The raised research questions point to further research topics which need to be studied to finally end up with an integrated pricing approach for spls.

References

Bayer J, Flege O, Knauber P, Laqua R, Muthig D, Schmid K, Widen T, DeBaud J (1999) Pulse: A methodology to develop software product lines. In: Proceedings of the Symposium on Software Reusability (SSR'99), Los Angeles.

Belleflamme P (2006) Versioning information goods. In: Illing G, Peitz M (eds.) Industrial organization and the digital economy. MIT Press, Cambridge.

Böckle G, Pohl K, v.d. Linden F (2005) A framework for software product line engineering. In: Pohl K, Böckle G, v.d. Linden F (eds.) Software product line engineering: Foundations, principles and techniques, Springer, Berlin.

Brinkkemper S (1996) Method engineering: Engineering of Information systems development methods. Inf. Soft. Tech. 38(4):275–280.

Buxmann P, Diefenbach H, Hess T (2008) Die Softwareindustrie: Ökonomische Prinzipien, Strategien, Perspektiven. Springer, Berlin.

Chastek GJ, McGregor JD (2009) Modeling variation in production plannig artifacts. In: Benavides D, Metzger A, Eisenecker U (eds.) 3th International Workshop on Variability Modeling of Software Intensive Systems, Universität Duisburg-Essen.

Clements P, Northrop L (2007) Software product lines: Practices and patterns, 6. Addison-Wesley, Boston.

Cohen S (1999) From product-line architectures to products. In: Moreira AMD, Demeyer S (eds.) Object-Oriented Technology, ECOOP'99, Springer, Berlin.

Czarnecki K (2005) Overview of generative software development. In: Banatre J, Fradet J, Giavitto JL, Michel O (eds.) Unconventional programming Paradigms, Springer, Heidelberg.

Czarnecki K, Eisenecker U (2000) Generative Programming: Methods, tools, and applications. Addison-Wesley, Boston.

Deneckere R, McAfee RP (1995) Damaged goods. Jour. Econ. Mngmt. Strt. 5(2):149–174.

Helfrich A, Schmid K, Herzwurm G (2006) Product management for software product lines: An unsolved problem?. Commun. ACM 49(12):66–67.

Helfrich A, Schmid K, Herzwurm G (2006a) Reconciling marketed and engineered software product lines. In: Proceedings of the 10th International Software Product Lines Conference (SPLC'06), IEEE Comp. Society, Los Alamitos.

Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS (1990) Feature-Oriented Domain Analysis (FODA) feasibility study, Tech. Report, Software Engineering Institute, Carnegie-Mellon University.

Kang KC, Kim S, Lee J, Kim K, Shin E, Huh M (1998) FORM: A Feature-Oriented Reuse Method with domain-specific reference architectures. An. Sof. Eng. 5(1):143–168.

Kang KC, Lee J, Donohoe P (2002) Feature-oriented product line engineering. IEEE Soft. 19(4):58–65.

Kang KC, Lee K, Lee J, Kim S (2003) Feature oriented product line software engineering: principles and guidelines. In: Itoh K, Satoshi K, Hirota K (eds.) Domain oriented systems development: perspectives and practices, Taylor & Francis, London.

Kotler P, Armstrong G (2006) Principles of marketing. Prentice Hall, Upper Saddle River.

Krueger CW (2006) New methods in software product line development. In: Proceedings of the 10th International Software Product Lines Conference (SPLC'06), IEEE Comp. Society, Los Alamitos.

Kuenne RE (1998) Price and nonprice rivalry in oligopoly: The integrated battleground. Macmillan, Basingstoke.

v.d. Linden FJ, Schmid K, Rommes E (2007) Software product lines in action: The best industrial practice in product line engineering. Springer, New York.

McAfee RP (2007) Pricing damaged goods. Econ. Open-Acc., Open-Ass. E-Jrnl 1(2).

Monroe KB (1990) Pricing: Making profitable decisions, 2. McGraw-Hill, New York.

Müller J (2009) Some economic aspects of software product lines. In: Ruhland J, Kirchner K (eds.) 10. interuniversitäres Doktorandenseminar Wirtschaftsinformatik, Jena Research Papers in Business and Economics, Universität Jena.

Nagle TT, Hogan JE (2006) The strategy and tactics of pricing: A guide to grow more profitabiliy, 4. Prentice Hall, Upper Saddle River.

Niehaus E, Pohl K, Böckle G (2005) Product management. In: Pohl K, Böckle G, v.d. Linden F (eds.) Software product line engineering: Foundations, principles, and techniques, Springer, Berlin.

Phlips L (1983) Economics of price discrimination: Four essays in applied price discrimination, Cambridge Univ. Press, Cambrigde.

Porter ME (1998) Competitive Strategy: Techniques for Analyzing Industries and Competitors, Free Press, New York.

Schmid K, Verlage M (2002) The economic impact of product line adoption and evolution. IEEE Softw. 19(4):50–57.

Schmid K (2002) A comprehensive product line scoping approach and its validation. In: Proceedings of the 24th International Conference on Software Engineering, New York.

Schmid K (2002a) A initial model of product line economics. In: Proceedings of the 4th International Workshop on Software Product Family Engineering PFE '01, Springer, London.

Sewerjuk D (2008) Pricing of software product lines. In: Bichler M, Hess T, Krcmar H, Lechner U, Matthes F, Picot A, Speitkamp B, Wolf P (eds.) Tagungsband Multikonferenz Wirtschaftsinformatik, GITO, Berlin.

Shapiro C, Varian H (1998) Information rules: A strategic guide to the network economy. Harvard Business School Press, Boston.

Vives X (2000) Oligopoly pricing: Old idea and new tools, MIT Press, Cambridge.

Whitey J (1996) Investment analysis of software assets for product lines. Tech. Report, Software Engineering Institute, Carnegie-Mellon University.

Wilson R (1993) Nonlinear pricing. Oxford Univ. Press, New York.