Results of the German Software Industry Survey 2013

12th August 2013

TECHNISCHE UNIVERSITÄT DARMSTADT

WIRTSCHAFTS INFORMATIK

Software-Cluster

Results of the German Software Industry Survey 2013

12th August 2013

Authors:

Anton Pussep
pussep@is.tu-darmstadt.de

Markus Schief
schief@is.tu-darmstadt.de

Tobias Weiblen
tobias.weiblen@unisg.ch

Timo Leimbach
timo.leimbach@isi.fraunhofer.de

Juhana Peltonen
juhana.peltonen@aalto.fi

Mikko Rönkkö
mikko.ronkko@aalto.fi

Prof. Dr. Peter Buxmann
buxmann@is.tu-darmstadt.de

Contents

1 Executive Summary

This is the second year that we execute the Software Industry Survey in Germany[1] and publish a report with the main results (see Pussep et al., 2012). As formulated in 2012, our goal is to investigate the current state of the German software industry on a yearly basis. Conclusions are made based on grounded data and empirical findings. We hope that this type of research will contribute to the work of both, practitioners and researchers.

As a public research institution we keep our respondents' data strictly confidential. We conduct this research in the context of the Software-Cluster[2], which is funded by the German Federal Ministry of Education and Research[3] under grant "01IC10S05". The authors take the responsibility for the contents.

Last year, we analyzed firm strategies and business models in the software industry. We found high growth for the German software and IT industry and even higher growth aspirations among software firms in our last year's sample. With regard to business models, we identified multiple relevant characteristics and found a high diversity across firms. In spite of the multifaceted business models, certain patterns prevailed, such as high shares of on-premise solutions (as opposed to on-demand solutions) and rising importance of mobile systems.

This year we further center our analyses around software firm's business models and strategies. Beyond the examination of business model characteristics, we focus on two additional important topics: Strategic Groups that emerge based on similar firm strategies and Software Ecosystems that focus on external resources complementing a firm's value offering and creation. Some of this year's results are the following:

- Despite the young age of firms in the industry overall and high numbers of small firms, large firms dominate the industry in terms of personnel and revenue. This accentuates high concentration within the industry and importance of large firms for the overall industry.

- With regard to business models, we found a high variety in firms' choices. However, on average, we found certain tendencies in our sample. For instance, software firms regard differentiation as more important than low-cost strategies. On the other hand, profitability and growth are two simultaneous goals. Further, traditional choices prevail, such as charging end-users directly, usage-independent pricing, and well-established platforms (servers, desktop/laptop). Though on-demand software, mobile, and cloud computing platforms are clearly gaining market shares, they remain of less importance today.

- As an additional entitity of analysis, we included strategic groups, which combine several business model characteristics to abstract groups of firms. Three strategic groups emerged from the data in our sample: on-premise standard software, on-demand standard software, and individual software. A performance analysis of these groups revealed that the most recent group of on-demand standard software is the least performant group as of today (in terms of revenue and risk).

- Our study also investigated collaborative structures, which are known as software ecosystems. Only about 30% of responding firms categorized themselves as independent, the remainder builds partnerships with other software vendors. These ecosystems are partially characterized by a strong lead firm ("hub") which is surrounded by smaller firms around its platform or product ("spokes"). For Germany, the most frequently named hubs were: Microsoft, Oracle, SAP, Google/Android and Apple. Spokes mainly are interested in the technology and reputation provided by their respective hub, while they try to differentiate from their hub through niche knowledge and strong customer relationships.

This report will hopefully assist practitioners, inform policy-makers, and support researchers with unique data. For practitioners working in or with the software industry, the report is further supplemented by various online materials which can be accessed through our website. Most importantly, we published an online tool to complement the application of this survey's results: The Business Model Wizard[4] allows to configure your software firm's business model and to benchmark the configuration against a database. By this means you can analyze similar business models and their performance indicators. We hope that this report as well as the Business Model Wizard support the optimization of your software business.

If you are interested in any further aspects of our research, please feel free to get in touch with us at any time. We are more than happy to answer further inquiries and we highly appreciate your valuable feedback.

[1] https://www.softwareindustrysurvey.de

[2] http://www.softwarecluster.org/en/

[3] http://www.bmbf.de/en/

[4] http://www.software-business-model.com

2 Industry Overview

In our report last year we concluded that the German software and IT industry is growing steadily, even though the overall performance is average when compared to other European countries. As in the previous year, we see steady growth in 2012. We further find empirical evidence that the software and IT industry is a young and dynamic industry, which is dominated by large players. However, there is also evidence that small and medium enterprises are gaining importance.

2.1 Industry Growth

Figure 2.1 show the growth rates of the software and IT industry in Germany in 2011 and 2012. The employment in this industry increases steadily. So do revenues, even though with much higher volatility. With growth rates well above 5%, we can conclude that the software and IT industry is growing well above average of the overall economy in Germany. With this in mind, we expect that the importance of software and IT will further increase for the German economy.

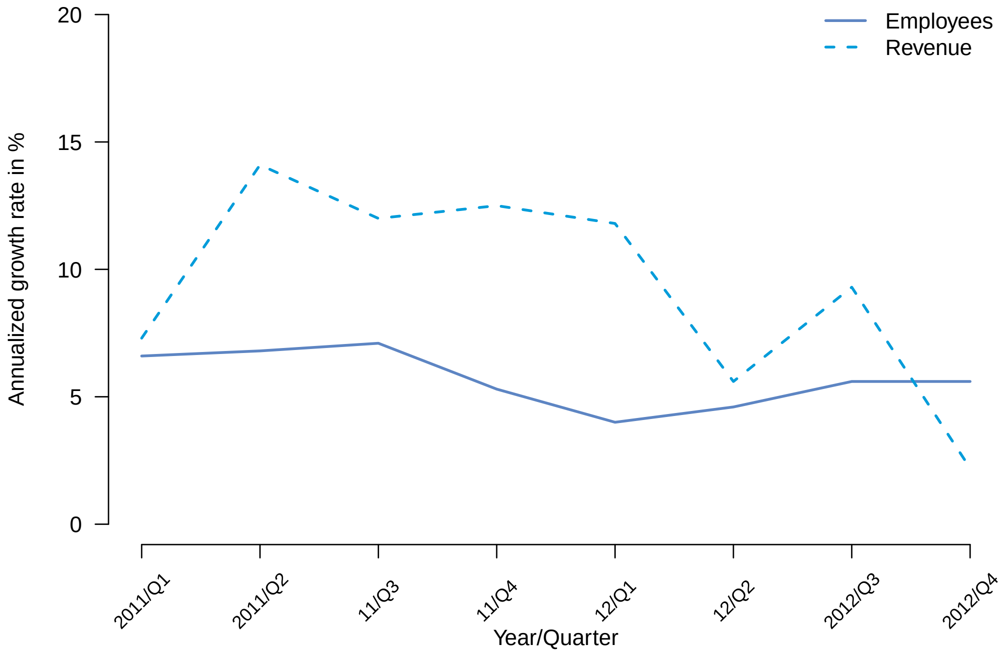


Figure 2.1.: Industry growth rates as measured by number of employees and revenue. Source: Eurostat, statistics for industry NACE R2 code 62 (computer programming and consultancy).

2.2 Industry Concentration

Knowing that the industry is gaining importance overall, we analyze whether large or small and medium firms are of higher importance for Germany. Clearly, this can impact the decisions of policy makers.

As Figure 2.2 indicates, a small group of large software and IT firms contributes a major share to the overall revenue and employment. On the other side, the group of small firms (up to 9 employees), which is by far the largest group in numbers, contributes a much smaller share to the overall revenue, employment, and gross value added.

According to Eurostat, in 2010, 20.276 firms (around 87%) employed 1-9 employees (including the owner). Only 171 (around 0.3%) employed more than 249 employees. Including the 1,147 firms employing 50-249 employees (1.99%), these large firms employ 52.69% of all employees working in this industry. The small firms of 9 employees or less account for only 22.38% of the overall employment. At the same time, these 2.4% of large and very large firms accounted for 68.7% of revenues and 65.6% of gross value added. The 87% of very small firms contributed only 12.51% (8.6 billion EUR) of revenues and 14.74% (0.34 billion EUR) of gross value added.

From these statistics, it becomes clear that the software and IT industry is highly concentrated. Looking at the development of the past years (Schleife and Leimbach, 2012, p: 15-16), we can see that the shares of small and very small firms have been increasing in terms of employment, revenue, and gross value added. Similar developments can be found in the group of medium firms (20-49 employees). These reduce the share of large and mostly very large firms. Overall, it appears that there is a tendency toward a stronger position of smaller firms and thus a halt to the tendency toward higher concentration.

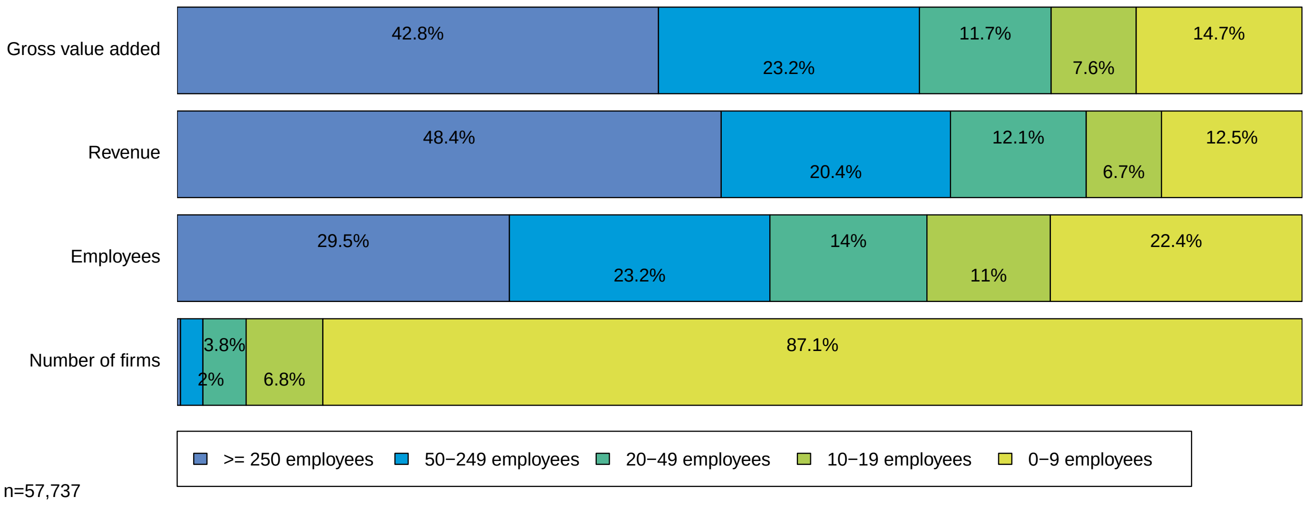


Figure 2.2.: Firm demography.

Clearly, these figures represent the broader IT industry and not only software firms which we target in our survey. However, we found similar concentrations in our sample for the software industry as illustrated by Figure 2.3. E.g., 50% of our respondents can be categorized as very small firms (1-9 employees). However, the share of large and medium firms is much higher in our sample. Though this might be a result in differences between the IT industry overall and software industry in particular, we also think that our survey is of particular interest to larger firms, thus leading to higher response rates across larger firms.

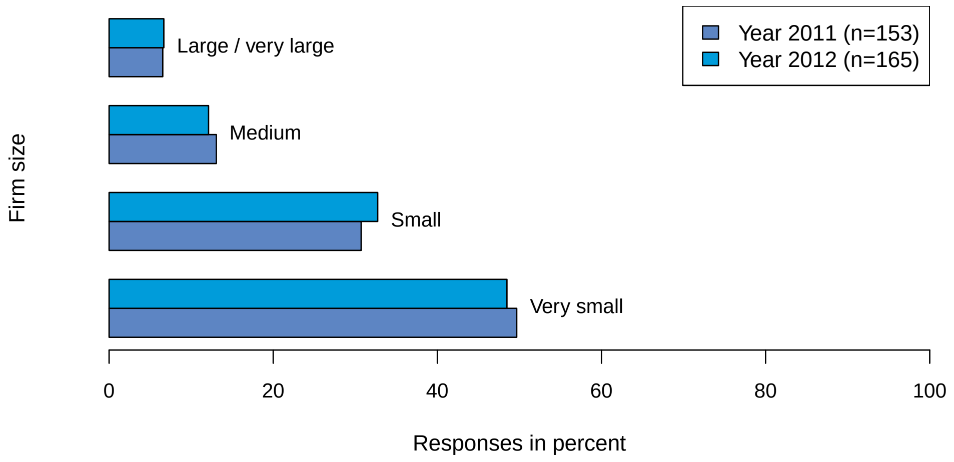


Figure 2.3.: Firm size measured by the number of employees.

Figure 2.4 does not confirm a halt to the tendency toward higher concentration in the German software industry. In our sample, small and large firms seem to grow faster than medium firms. This result is confirmed in terms of both, personnel and revenue growth. Given these statistics, we cannot conclude how concentration in the software industry will develop further. From these figures we may further conclude that medium-sized firms had difficulties to keep up with the average industry growth from 2011 to 2012.

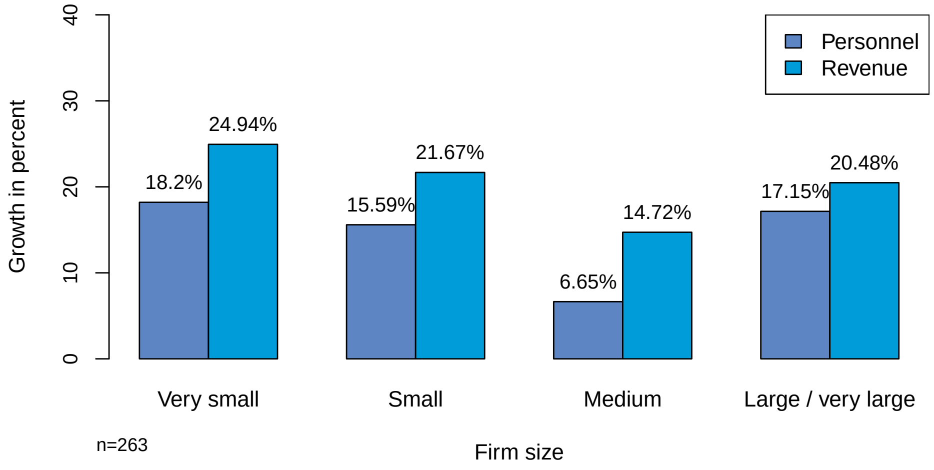


Figure 2.4.: Personnel and revenue growth by firm size (measured by the number of employees).

2.3 Industry Dynamics

Knowing that the major share of software firms in our sample is made up of small or very small firms, the proportions could be explained by firm age as shown in Figure 2.5. The median age in our sample is 11 years, thus more than half of the firms have been founded after the dot-com bubble. This further indicates that industry dynamics are high, leading to high numbers of new firms in the market, but also high numbers of firms disappearing from the market.

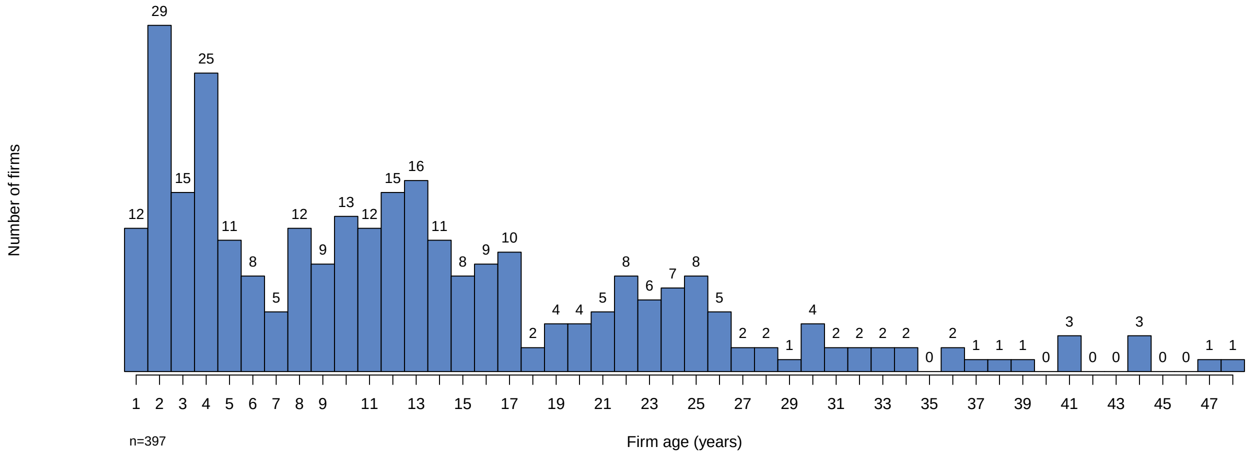


Figure 2.5.: Firm age.

2.4 Firm Types in the Software Industry

In our survey, we asked the participants to classify their firm according to a given scheme. The results are shown in Figure 2.6. As last year, the majority can be termed software product firms. This group of firms is closely followed by two other large groups, being individual software development firms and implementation firms. On the other hands, it appears that software resellers, IT-related consulting firms and embedded software firms make up a small share of our sample. As a result, our conclusions do not necessarily apply to the hardly represented groups, as the number of participants is too low for meaningful statistical analyses.

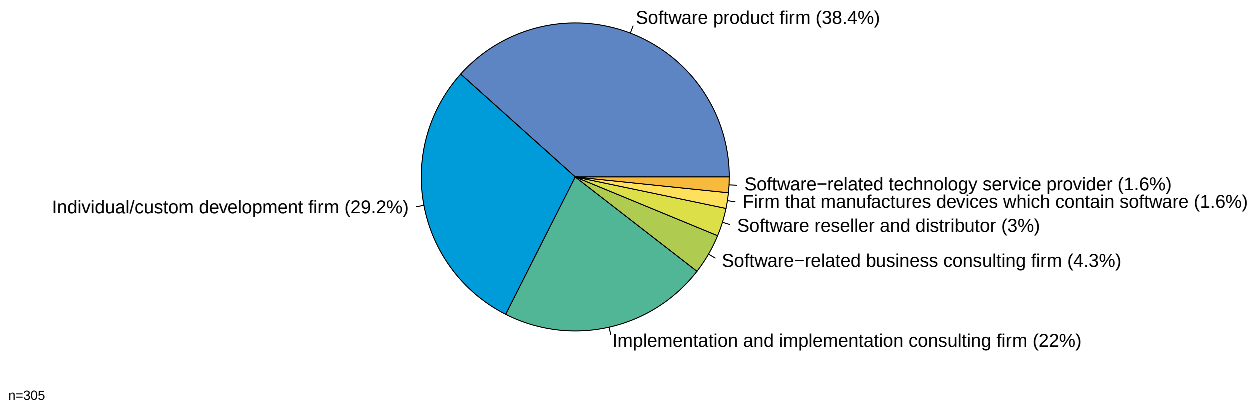


Figure 2.6.: Firm types in the software industry.

2.5 Firm Profitability

Finally, we turn to the question whether software firms are profitable. Figure 2.7 shows the operating margins for all firms as well as for each firm size class. We can see that the median operating margin is 10-20%. The average number (not shown) is closer to 10-20% than to 0-10%, thus pointing in the very same direction. There seems to be no relationship between firm size and firm profitability, as the margin distributions across firm sizes look much alike, with most firm generating margins ranging from 0-20%. From that, we conclude there seems to be no general relationshop between firm size and profitability in the software industry. However, we do think that size effects can be found when looking at more homogenous sub-samples of the software industry. E.g., we would expect to find differences in profitablity for software product firms because of network effects.

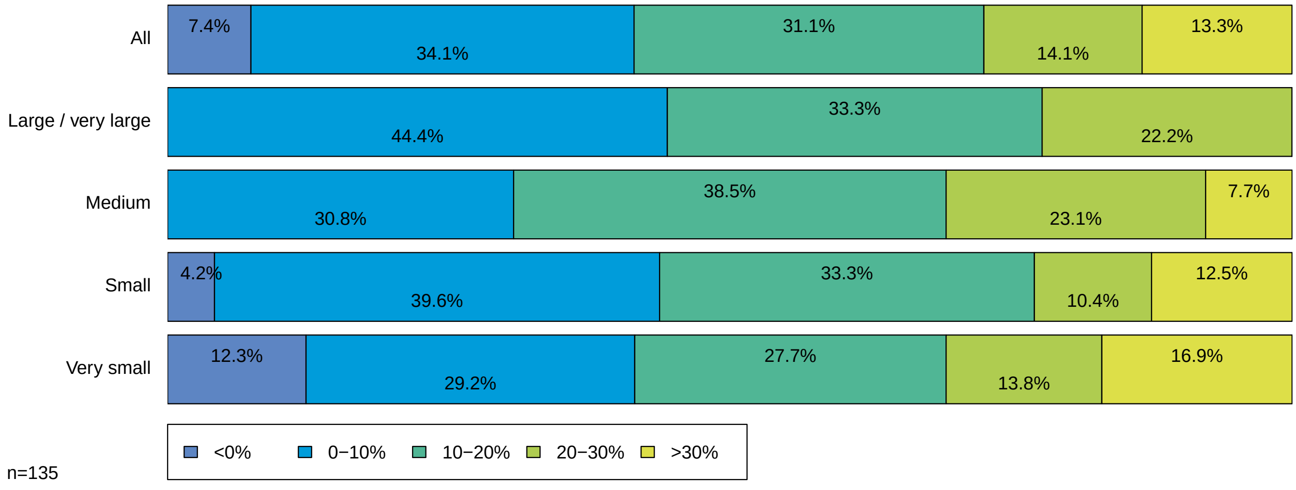


Figure 2.7.: Firm profitability (measured by operating profit margin). Overall figures and figures by firm size (measured by number of personnel).

2.6 Excursus: Software Industry in Finland

The Finnish Software Industry Survey[1] was run for the 16th time in Spring 2013, and its results were published on June the 11th. The downsizing of Nokia's software development makes Finland a special case in the European context. Yet,

[1] http://www.softwareindustrysurvey.org/

the Finnish software and IT services sector (Nace rev. 2 code 62) grew by 5.8% percent in 2012. According to Statistics Finland, the total revenues were 6.2 billion EUR in 2012. Though this is less than a tenth of Germany's volume of 68.7 billion EUR, it represents 3.2% of Finland's GDP compared to 2.6% of Germany.

While the heavy layoffs of Nokia's subcontractors still dragged the overall growth of the Finnish software industry, the situation of small and medium-sized enterprises seems to have improved from the previous year. Overall growth was driven mostly by a few of successful game companies (particularly Rovio and Supercell). In 2012, Finnish game industry generated approximately 250 million EUR in revenues, and, the figure may reach 600-800 million EUR in 2013[2]. This would mean that perhaps over 10% of Finland's IT service sector's revenues come from games.

One of the major challenges of the industry is that promising young companies rarely develop into steadily growing and employing large companies. In recent years, the number of software companies on the Helsinki Stock Exchange has in fact decreased, even though the founding rate of software firms has remained constant or even slightly increased.

The degree of internationalization among Finnish firms achieved two milestones in 2012. For the first time in the history of the survey more than half of the responding companies had international revenues. Moreover, more than half of the firms that did not have international revenues stated that they are either planning or had previously attempted internationalization. For the first time, Russia and Eastern Europe surpassed North America as a target market. Although the number of entries shows that Russia and Eastern Europe are increasing, the North American markets still generate more revenue. Scandinavia and Western Europe remained the most common target market by a large margin.

2.6.1 Software Industry Restructuring Continues

While Nokia's staff reductions are today much less in the headlines than in 2012, the restructuring is still ongoing. Open job positions in the software industry dropped from around three hundred to around two hundred in 2012, but started to increase slightly in 2013. In 2012, small and medium-sized software companies recruited about 350 people that were previously employed by Nokia or any of its subcontractors. In addition, companies like Intel, Samsung and Huawei have been expanding or setting up new R&D centers in Finland to grab some of the IT experts that have been freed up from the Nokia cluster.

2.6.2 Cloud Platforms Passed Mobile Platforms in Software Development

Software development in Finnish software SMEs is rapidly moved to take advantage of cloud computing platforms: 47% of software companies announced to develop software that takes advantage of cloud computing platforms. For comparison, this number exceeds that of mobile for the first time (38%), which has traditionally been a Finnish stronghold.

The results suggest that Finnish software companies take advantage of cloud computing both to enhance existing business and to exploit new business opportunities. One differentiating factor between firms that use cloud computing and those that do not is that firms that develop for the cloud platforms seem to be more growth oriented.

The survey also shows that the race between smart phones development platforms is genuinely a competition between three: Android and iOS, and Windows Phone with 20%, 19% and 15% shares, respectively. Symbian has fallen to below five per cent, and expected to decrease further.

The data from the Finnish survey will be analyzed further, and factors limiting firm growth and funding will be published in the autumn of 2013.

[2] http://www.neogames.fi/tietoa-toimialasta/

3 Business Models: The Software Firm's Business DNA

This section analyzes business models of software firms. The concept of business models can be considered as a rather young field that has been rising over the past decade in research and practice. In the software industry, the choice of the right business model is a crucial success factor for sustainable business success. Firms like Google or Apple have quickly become the world's most valuable brands and each has dramatically demonstrated that every technological innovation also requires an appropriate business model. Therefore, business models in the software industry and their underlying mechanisms are examined again in this year's survey.

While the term business model is broadly used in entrepreneurial practice, the definition, nature, and structure of business models is still an object of debate among researchers (Burkhart et al., 2011). As terms such as strategy, business model, and revenue model are often used interchangeably, confusion in terminology is common. In the following presentation of results, we build upon an business model concept by Schief and Buxmann (2012) that is specific to software firms.

The overall business model can be divided into five groups as shown in Figure 3.1. Each group in turn consists of business model components. For each component, a firm has several options on how to instantiate the given component. E.g., within the group *Revenue*, a firm has three options to instantiate the component *Revenue source*: direct revenue generation from customers, through advertising, or commission.

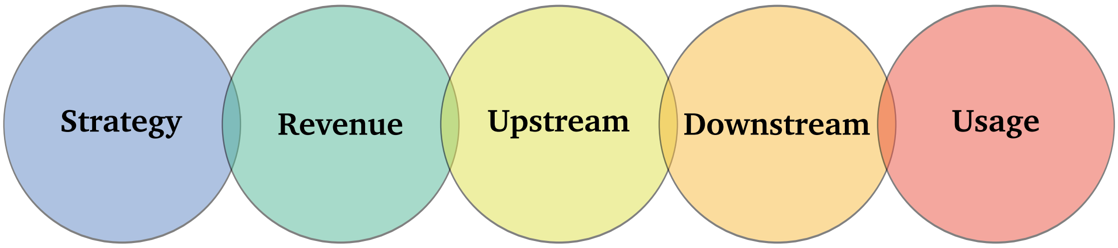
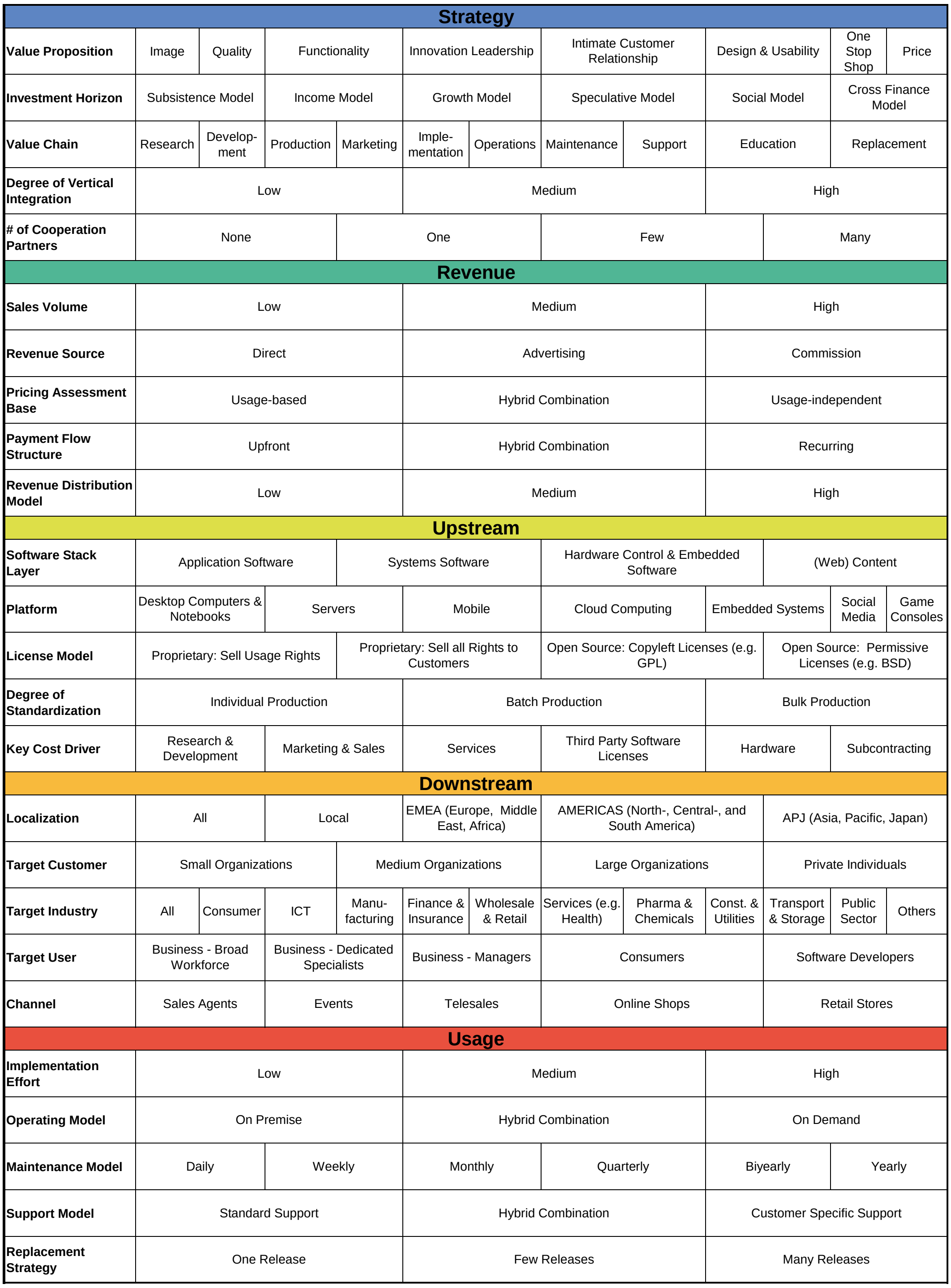


Figure 3.1.: Five groups within the business model concept.

Each group summarizes multiple cohesive business model components: The group *strategy* describes a firm's strategic decisions, the group *revenue* deals with the pricing model and financial flows, *upstream* covers components that relate to the definition and development of a product or service, *downstream* considers the target market and the distribution channel, and *usage*, finally, consists of services that are necessary during the usage of a software solution. The detailed business model concept including the five groups and their 25 components is shown in Figure 3.2.

The following list provides a detailed description of all components' choice options (Schief, 2013):

1. Group: Strategy
    - Value Proposition: This component describes the competitive advantage of a firm's offering. The choice options represent options to achieve a differentiation or cost leadership position. Image deals with the perceived value that customers attribute to firms. Quality stands for high consistency and dependability. Functionality covers the breadths and depths of available features. Innovation Leadership can be achieved through new and disruptive offerings. Intimate Customer Relationship describes the relationship intensity between a firm and its customers. Design & Usability depend on the ease of use, intuition, and visual attractiveness of a solution. One Stop Shops offer customers bundled end-to-end solutions comprising all necessary components (e.g. hardware, software, and services). Finally, Price implies to offer a solution at lowest cost. This option is often realized by focusing on cost reduction and efficiency.
    - Investment Horizon: This component deals with the business model's strategic time horizon. The goal of a Subsistence Model is to survive and meet basic financial obligations. When employing an Income Model a firm invests to the point that the business is able to generate an ongoing and stable income stream for the principals. A Growth Model is an attempt to grow the value of the firm to the point that it eventually generates a major capital gain for investors. The timeframe of a Speculative Model is shorter and the objective is to demonstrate venture potential before selling out. The Social Model is applied by organizations that do not follow any profit goal. Finally, the Cross Finance Model represents a subsidy to another line of business (e.g. to support a complimentary offering).



Strategy												
Value Proposition	Image	Quality	Functionality	Innovation Leadership	Intimate Customer Relationship	Design & Usability	One Stop Shop	Price				
Investment Horizon	Subsistence Model	Income Model	Growth Model	Speculative Model	Social Model	Cross Finance Model						
Value Chain	Research	Develop-ment	Production	Marketing	Imple-mentation	Operations	Maintenance	Support	Education	Replacement		
Degree of Vertical Integration	Low	Medium	High									
# of Cooperation Partners	None	One	Few	Many								
Revenue												
Sales Volume	Low	Medium	High									
Revenue Source	Direct	Advertising	Commission									
Pricing Assessment Base	Usage-based	Hybrid Combination	Usage-independent									
Payment Flow Structure	Upfront	Hybrid Combination	Recurring									
Revenue Distribution Model	Low	Medium	High									
Upstream												
Software Stack Layer	Application Software	Systems Software	Hardware Control & Embedded Software	(Web) Content								
Platform	Desktop Computers & Notebooks	Servers	Mobile	Cloud Computing	Embedded Systems	Social Media	Game Consoles					
License Model	Proprietary: Sell Usage Rights	Proprietary: Sell all Rights to Customers	Open Source: Copyleft Licenses (e.g. GPL)	Open Source: Permissive Licenses (e.g. BSD)								
Degree of Standardization	Individual Production	Batch Production	Bulk Production									
Key Cost Driver	Research & Development	Marketing & Sales	Services	Third Party Software Licenses	Hardware	Subcontracting						
Downstream												
Localization	All	Local	EMEA (Europe, Middle East, Africa)	AMERICAS (North-, Central-, and South America)	APJ (Asia, Pacific, Japan)							
Target Customer	Small Organizations	Medium Organizations	Large Organizations	Private Individuals								
Target Industry	All	Consumer	ICT	Manu-facturing	Finance & Insurance	Wholesale & Retail	Services (e.g. Health)	Pharma & Chemicals	Const. & Utilities	Transport & Storage	Public Sector	Others
Target User	Business - Broad Workforce	Business - Dedicated Specialists	Business - Managers	Consumers	Software Developers							
Channel	Sales Agents	Events	Telesales	Online Shops	Retail Stores							
Usage												
Implementation Effort	Low	Medium	High									
Operating Model	On Premise	Hybrid Combination	On Demand									
Maintenance Model	Daily	Weekly	Monthly	Quarterly	Biyearly	Yearly						
Support Model	Standard Support	Hybrid Combination	Customer Specific Support									
Replacement Strategy	One Release	Few Releases	Many Releases									

Figure 3.2.: Components of a software firm's business model. Source: Schief (2013).

- Value Chain: This element summarizes the main value chain activities that a firm may cover.

  Research: This activity comprises fundamental product research. A product vision is developed and fundamental algorithms are researched. Major technologies and subsystems are selected. A first proof of concept is provided through a prototype or analysis of algorithms, technologies, and subsystems. The result is a product idea, algorithm or proof of concept. Unlike in the following activities, no code is created here that becomes part of the actual product.

  Development: This activity deals with the actual software development process. Based on requirements, a software design is created. The entire system is decomposed into subsystems. Subsystems are programmed and tested separately, before they are integrated and tested as a combined system. The user documen-tation is created and the product is compiled to an executable and versioned product. The result is an executable version of the product.

  Production: Within assembly, software and respective documentation are bundled to one package. The assembled software package is printed to a physical medium and the documentation is printed on paper. In packaging the physical product artifacts are packaged in a physical package. The result is a product with all attributed artifacts being ready for shipment.

  Marketing: Providing a means by which buyers can purchase the product and inducing them to do so, such as sales and promotion. The result is the readily marketed product in the marketplace, such that potential customers are aware of the product and the product is available for purchase.

  Implementation: The installation comprises the transmission of the packaged binaries to the customer's information system. Moreover, it ensures that the binaries can be executed without runtime errors. Configuration allows the setting of software parameters and software modifications according to the customer's needs. Finally, adaptations can be performed that modify or enhance the functionality of the software product and employ business process changes.

  Operations: The operations activity ensures the execution and management of a product on an information system during actual usage by customers. The system behavior can be analyzed and supervised through monitoring. To minimize dam-ages through data loss, regular data back-ups need to be planned, run, and ad-ministered. Finally, the information system needs to be upgraded to new releases during its lifecycle.

  Maintenance: Likewise to development, but the focus is on bug fixing and enhancing an existing product, whereas the activity development aims at the creation of a new product. Within maintenance, disruptive changes are not allowed. Instead, incremental changes are made by the producer to an existing product in the marketplace.

  Support: Support can be differentiated in primary and development support. While the first sub-activity deals with the support of users, the second activity relies on deep technical knowledge and implies code reviews.

  Education: Training of users and third party firms. In addition, certifications attest users and third party firms a certain degree of seniority in the handling of a software product.

  Replacement: First, replacement deals with the decision if the product (once it becomes outdated and reaches the end of its lifecycle) shall be replaced by an alternative system. If the decision for an alternative is made, data needs to be mi-grated from the legacy to the new system. Subsequently, the legacy system is shut-down. A seamless transition to the new system is the main target at this stage. After the irrevocable data destruction of confidential information, the shut-down activity is completed.

- Degree of Vertical Integration: Each value chain activity can be either performed internally or externally. The degree of vertical integration measures how many value chain activities are performed within the boundaries of a single firm compared to the overall set of required value chain activities. The more value chain activities are outsourced to partners, the lower is the degree of vertical integration. The choice options Low, Medium, and High represent a qualitative judgment of the degree of vertical integration.

- Number of Cooperation Partners: This component deals with the number of partners that a firm cooperates with. Four main choice options are considered. None represents firms that do not rely on any other company. These firms usually have a very high degree of vertical integration. One stands for firms that mainly rely on one other company. For example, software consulting firms offering implementation projects based on another software vendor's product. Few implies that a firm enhances the number of cooperation partners slightly and hence outsources some value chain activities to partners. Finally, firms can rely on Many cooperation partners. For example, if a firm develops a software infrastructure platform (e.g. mobile operating system) that is used by various other firms for application development.

2. Group: Revenue
    - Sales Volumes: This component describes the number of sold solutions. Usually, this number refers to the number of installations as one customer may have more than one installation. The choice options Low, Medium, and High represent qualitative judgments.
    - Revenue Source: This component investigates who finally pays for a solution. Direct means that the user pays for the solution. Advertising stands for revenues that stem from third parties instead of the user. For instance, Google Inc. does usually not charge product users. Instead they sell advertisement space to third parties. Finally, Commission refers to cases, where firms offer brokering services (e.g. Ebay Inc.) and receive a percentage of the transaction value.
    - Pricing Assessment Base: This component explores how prices are defined. Two dominant pricing schemes can be differentiated. Usage-based pricing schemes calculate the price based on the actual usage of a product (e.g. used storage or CPU power). Usage-independent pricing schemes, in contrast, do not reflect the actual usage of a solution (e.g. number of named users). Hybrid Combination is a mixed pricing scheme consisting of a usage-based and usage-independent pricing assessment base.
    - Payment Flow Structure: This component deals with the point in time when customers pay for a solution. Upfront payments imply that a customer pays an upfront license fee before using the product. Recurring payments, in contrast, stand for subscription fees. Customers then pay on a continuous basis. Hybrid Combination represents a mixed payment flow structure. For instance, if a vendor charges initial license fees and recurring fees for maintenance and support.
    - Revenue Distribution Model: This component stands for the percentage of revenues that is shared with stakeholders. The choice options Low, Medium, and High represent qualitative judgments. For example, mobile application developers usually need to share their revenue with the provider of the mobile application marketplace.

3. Group: Upstream
    - Software Stack Layer: This component classifies software based on a software stack concept. Application Software is designed to help users to perform specific tasks (e.g. ERP, accounting, office, media, and games). Systems Software is designed to integrate software and information systems (e.g. operating systems, middleware, and security). Hardware Control & Embedded Software stands for software that is bundled with hardware (e.g. firmware). (Web) Content represents offerings that focus on the information content rather than on the software itself (e.g. offer a library of IBAN banking numbers).
    - Platform: This component investigates the technical platform of a solution. Desktop Computers & Notebooks represent the traditional platform for client software installations. Further, software can be installed on Servers (e.g. mainframes). Today, Mobile becomes another increasingly important platform. Software then runs on the mobile devices. Cloud Computing (e.g. Force.com) refers to centralized servers that allow access through the internet. Besides, software can be installed on dedicated hardware devices. For instance, firmware runs on Embedded Systems. Social Media (e.g. Facebook Inc.) can be considered as a further platform that firms use as platform for software development. Finally, Game Consoles (e.g. Microsoft Xbox) build a common platform for application software in the entertainment sector.
    - License Model: This component describes the legal regulations associated with the software code. The choice options can be differentiated in two proprietary and two open source license models. Sell Usage Rights means that a firm sells the usage rights based on a proprietary license (customers may use the software without revealing the source code). Sell all Rights to Customers implies to sell all software rights (e.g. in custom development projects the usage right and source code may be fully transferred to the customer). Copyleft Licenses describe licenses where the developed software may not be labeled to another license type. In contrast, Permissive Licenses allow developing software that is published under a different license type.
    - Degree of Standardization: This component analyzes if a firm sells a highly standardized solution or a highly customer-specific one. Individual Production refers to tailor-made offerings that require knowledge on processes and technology interfaces being specific to each customer. Batch Production means that a company can reuse the same solution for a few customers. Finally, Bulk Production describes a solution that can be reused by any customer.
    - Key Cost Driver: This component asks for the dominating cost drivers. The choice options are grouped into two areas. While the first three choice options refer to personnel costs, the latter three cover purchase costs from third parties. Research & Development (i.e. developing the software), Marketing & Sales (i.e. salary of sales people), and Services (i.e. providing support, implementation, maintenance, and operations services)

are costs that result from the spending on personnel covering the respective value chain activities. In contrast, Third Party Software Licenses (e.g. software reseller), Hardware (if a firm buys and then resells hardware with software enhancements), and Sub-contracting (if a firm sells storage capacity while the datacenter is outsourced) are choice options that result in purchasing costs.

4. Group: Downstream

   - Localization: This component deals with the geographic areas that a firm addresses. All addresses a global market, Local stands for domestic markets, EMEA represents Europe, Middle East, and Africa, AMERICAS covers North-, Central-, and South America, and APJ denotes Asia, Pacific, and Japan.

   - Target Customer: This component explores the customer size. Small Organizations employ up to 50 employees. Medium Organizations range between 51 and 250 employees. Large Organizations employ more than 250 employees. Finally, Private Individuals are single persons buying the software.

   - Target Industry: This component describes the target industry that a software firm may address. All stands for firms that sell horizontal solutions (e.g. leave request systems) that can be applied in any industry Consumer represents solutions that are not considered for business, but for consumers. Nine further industries (Information and Communication Technology (ICT) Manufacturing, Finance & Insurance, Wholesale & Retail, Ser-vices (e.g. Health), Pharmaceuticals & Chemicals, Construction & Utilities, Transport & Storage, Public Sector) are explicitly listed as they are common target markets (see Section 3.3.5). Finally, Others covers all further industries that are not listed explicitly.

   - Target User: This component describes the type of users a solution is designed for. For business users, three types are differentiated: Broad Workforce (e.g. travel reimbursement), Dedicated Specialists (e.g. controlling, graphics), and Managers (e.g. dashboards). Further, Consumers use the software for personal use and Software Developers use the software to develop own software.

   - Channel: This component explores the sales channel types used to address customers. Sales Agents personally sell the solution to customers. Events can be used to attract a broad audience. Telesales still allows personal contact, while reducing travel costs. Online Shops stands for Internet sales channels. Retail Stores are shops offering direct customer contact.

5. Group: Usage

   - Implementation Effort: This component deals with the effort for product installation and configuration. The choice options Low, Medium, and High represent qualitative judgments. For example, implementation efforts can be considered low if the software installs quickly without further need for action before using the software.

   - Operating Model: This component analyzes how a software solution is deployed. Two main deployment models are offered as choice options. On Premise implies an installation and execution of the software on local systems at the customer side. On Demand, in contrast, stands for installation and execution on a central hosting platform allowing user access via Internet. The solution provider then also takes care for needed activities (e.g. backups) during software usage. Hybrid Combination represents a mixed deployment model. For instance, private cloud solutions centralize the deployment for one customer on a central cloud server.

   - Maintenance Model: This component investigates the release frequency of a software solution. In contrast to patches and hot fixes, a release stands for a major update of a product's software version. The choice options describe the frequency of new releases. Predefined intervals are Daily, Weekly, Monthly, Quarterly, Biyearly, and Yearly.

   - Support Model: This component explores what kind of support is needed by customers. It mainly depends on the type of support contract. Standard Support implies a one size fits all support offering. In contrast, Customer Specific Support means that each customer has a highly individual support contract. For instance, support offerings are often differentiated with respect to the underlying service level agreements (SLAs). Hybrid Combinations cover the support options in between this continuum.

   - Replacement Strategy: This component deals with the number of available product releases at a time. One Release implies that all customers are running on the same release version. Few Releases represent some major release versions that customers are using. Many Releases, finally, refers to a larger number of release versions that are applied by customers.

3.1 Strategic Aspects of Business Models

The first group puts a focus on strategic characteristics of a software business model by investigating the following five key components:

3.1.1 Value Proposition

In terms of value propostion, two main strategies are differentiated and examined accordingly, namely differentation and cost leadership strategies. Figure 3.5 shows that all seven differentation strategies achieve high importance values. Thereof quality, seems to be the moste important differentation strategy. Interestingly, research receives the comparably lowest level of importance.

With respect to cost leadership strategies, six aspects are investigated (see Figure 3.4). Thereof, economies of scale and scope as well as minimizing costs in general receive the highest values. Lower prices than competitors and specific cost groups such as advertising, development are not that important.

Notably, when comparing both strategy types, differentiation and cost leadership, it turns out that differentation strategies are considered to be far more important than cost leadership strategies. Thus, firms in the software industry strive on differentiating their offerings at reasonable costs instead of purely focusing on commodity products at low costs.

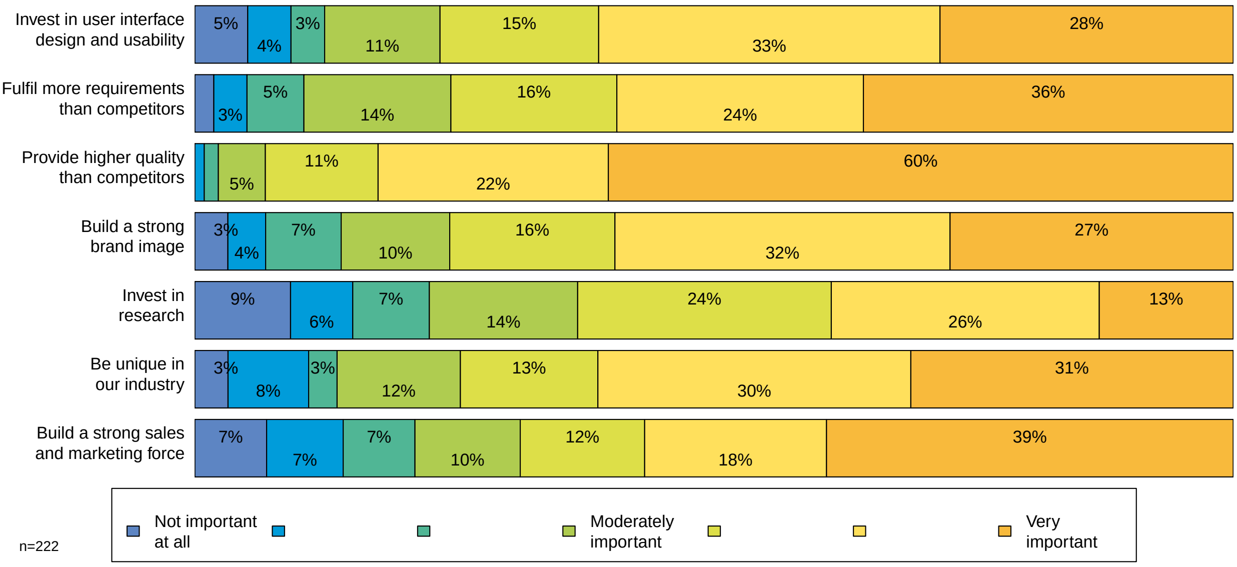


Figure 3.3.: Differentation strategies as value proposition.

3.1.2 Investment Horizon

In this study we approximate different investment horizons by examining the balance of growth and profit aspirations. The results accentuate the importance of both strategies namely high growth and profit rates. Figure 3.5 shows the distribution of six related questions. Interestingly, firms follow both strategies without sacrificing one for the other.

3.1.3 Value Chain

Next, the ten value chain activities are investigated with respect to their relative importance. Participants were asked to choose the three most important value chain activities. The distribution in Figure 3.6 illustrates that development is by far the most important activity according to more than 80% of respondents. In addition, implementation, maintenance, and support yield strong results. In contrast, replacement and production are only of minor importance.

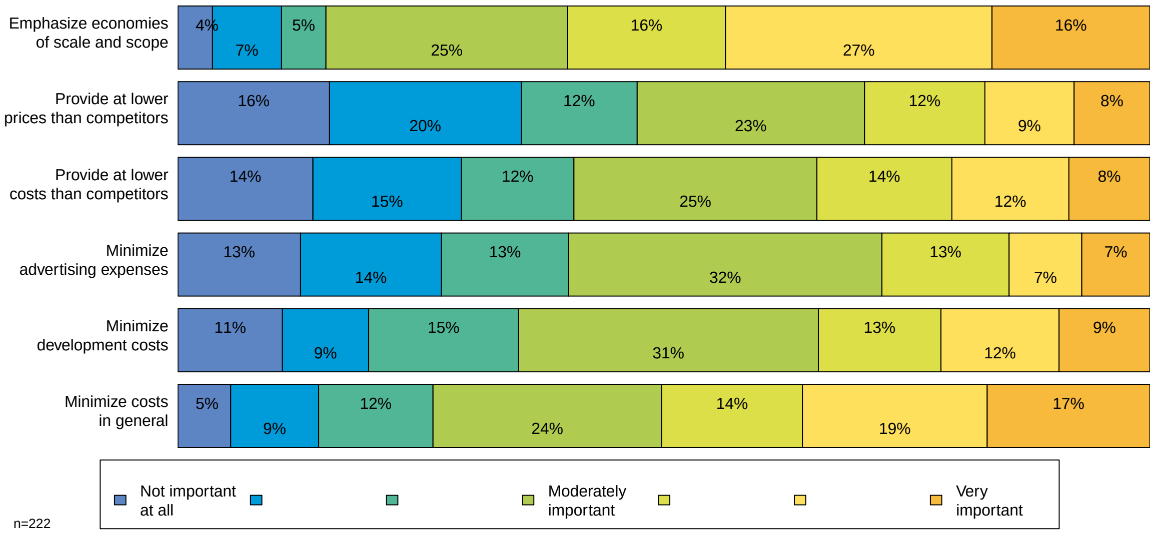


Figure 3.4.: Cost leadership strategies as value proposition.

3.1.4 Degree of Vertical Integration

Degree of vertical integration measures how many value is generated within a company itself. In this light, we investigate the importance of personnel costs (approximating the value generated inhouse) and purchase costs (approximating the value generated by others). Figure 3.7 shows that personnel costs are far more important than purchase costs. This result confirms the findings of last year's German Software Industry Survey concluding that the degree of vertical integration is comparably high to other industries (such as the automobile sector).

3.1.5 Cooperation Partners

The final strategic component deals with cooperation. In this regard, three aspects are investigated. Firstly, the number of cooperation partners is analyzed. Figure 3.8 depicts that most firms maintain many cooperations with other firms. Secondly, firms also engage in considerable exchange, sharing, or co-development with other firms. Nevertheless, the cooperation intensity values are lower than the cooperation number results. Finally, cooperations are mostly considered to be an important part of the firm strategy. Thus, overall, cooperation seems to be an important part of software firms' strategies.

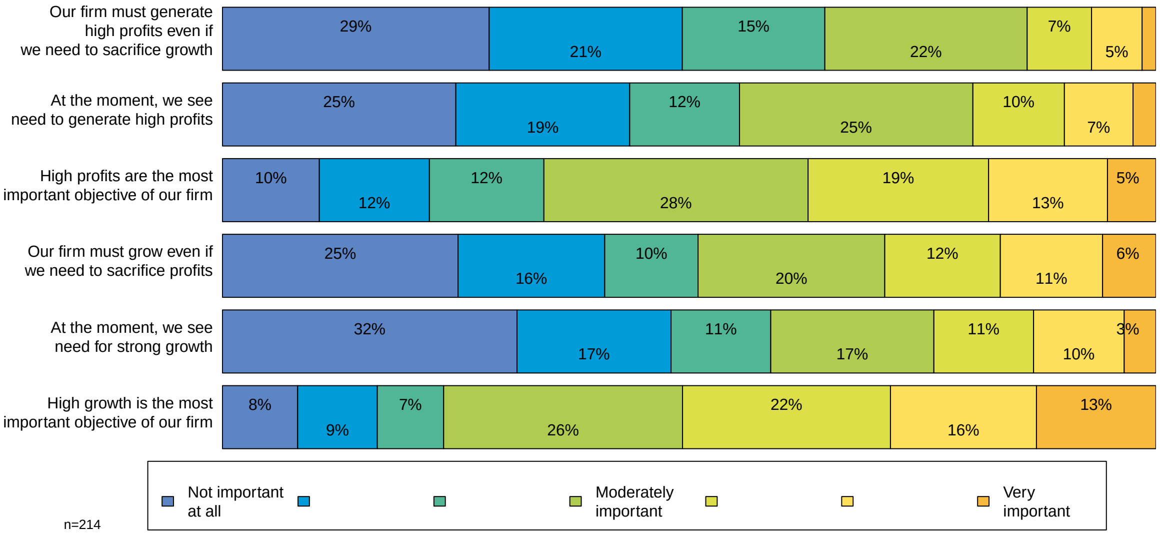


Figure 3.5.: Growth and profitability goals as investment horizon.

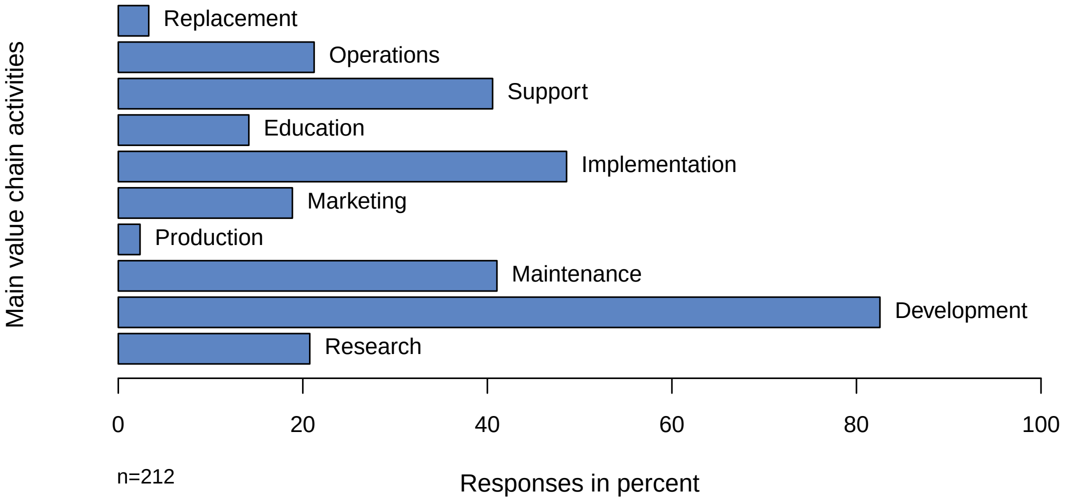


Figure 3.6.: Relevance of value chain activities.

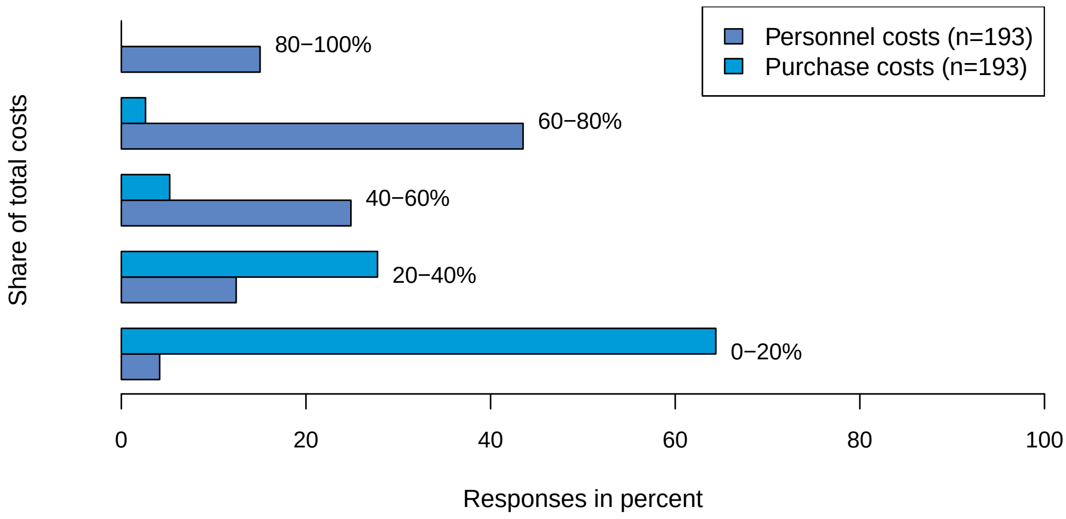


Figure 3.7.: Degree of Vertical Integration.

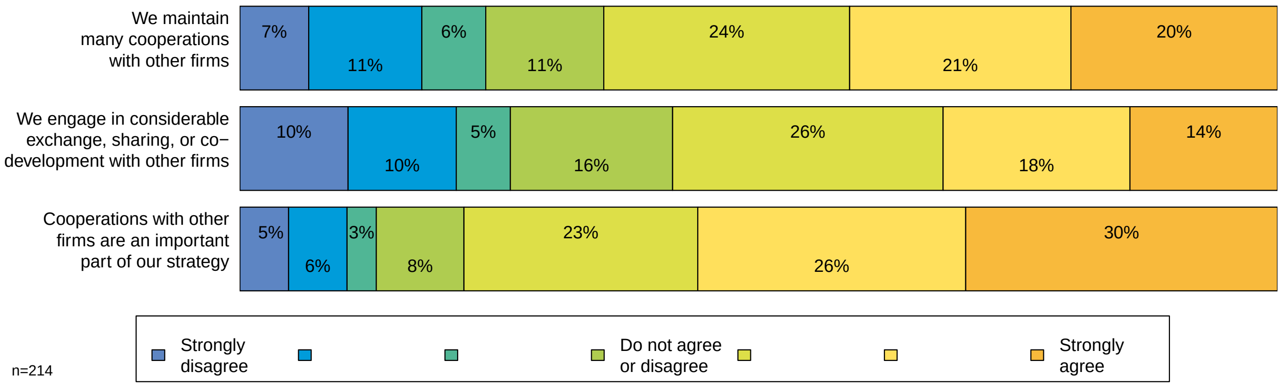


Figure 3.8.: Cooperation Partners.

3.2 Revenue Models

One fundamental dimension of business models deals with revenues. In terms of revenue, multiple aspects need to be considered. In our study we focus on five revenue and pricing components:

3.2.1 Sales Volume

This component investigates the firms' absolute sales volumes compared to their compeitors. Figure 3.9 illustrates that the distribution is slightly skewed to the left. Accordingly, more companies judge their sales volumes slightly lower compared to their competitors. A rationale behind this finding may refer to the sample structure being dominated by very small and small software firms. If the number of small firms is far higher than the number of big firms, more firms judge their absolute sales volumes comparably low.

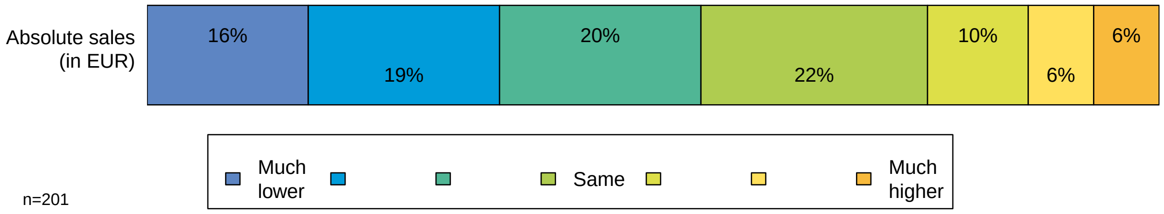


Figure 3.9.: Relative Sales Volume.

3.2.2 Revenue Source and Pricing Model

Beyond sales volumes it is interesting to examine, how revenue is earned and based on which pricing models. In this light, we analyze a software firm's revenue source, its structure of payment flows and its pricing assessment base. Figure 3.10 shows the results of all three aspects across all survey firms. For each component, the respondents could choose which of the two given options applies best.

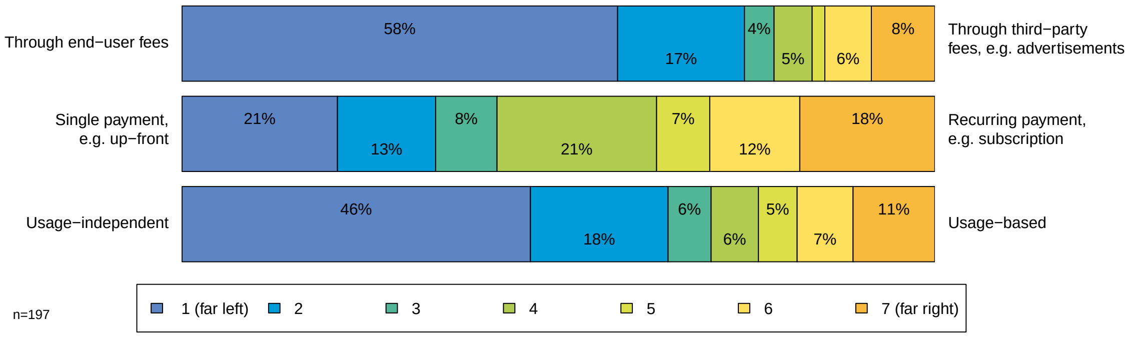


Figure 3.10.: Revenue Source and Pricing Model.

With respect to the software firms' revenue sources, the majority of firms yield their revenues through end-user fees. Only less than 10% mainly yield their revenues through third-party fees. Notably, the results are comparably stable to the fidings of our last year's German Software Industry Survey. This division of revenue sources means that German software firms still predominantly tend to rely on the well-established revenue source instead of exploring new revenue sources such as advertising. In the light of firms such as Google and Facebook, which have demonstrated how to make use of third party payments as primary source of revenue, this seems to be challenging for most other software firms.

In terms of payment flow structure, the division of results is more equally distributed. Though single payments are still applied as the dominating payment flow structure, roughly one third of the respondents mainly refer to recurring payments. Notably, more than half of the respondents report that they follow hybrid models combining initial upfront payments with recurring payments. For instance, customer initially pay for the license and pay continuously for maintenance and support services. It remains of interest how this distribution will evolve over time with an industry expecting rising software-as-a-services (SaaS) offerings which are typically associated with recurring payments. Nevertheless, compared to last year's German Software Industry Survey the increase in recurring payment schemes is comparably small.

With respect to the pricing assessment basis about half of the sample firms charge usage-independent prices. Pure usage-based pricing strategies are only followed by 11% of all firms. These results go in line with the findings of last year's German Software Industry Survey and Lehmann and Buxmann (2009). They report that only 14% of software vendors prefer a usage-dependent pricing assessment base. This number is expected to rise with the increasing number of software-as-a-service offerings. SaaS solutions being operated by the provider allow easier implementation of usage-based pricing strategies. However, the share of usage-dependent pricing remains below expectations.

All in all, the majority of revenue sources and pricing models still follows well-established strategies. Software firms tend to yield revenues through end-users and charge single payments based on usage-independent prices. Nevertheless, revenue and pricing strategies may evolve in the light of the rising trends such as software-as-a-service.

3.2.3 Revenue Distribution Model

Finally, the revenue distribution model investigates to which extent revenues are shared with other partners. Revenue sharing models are particularly relevant to platform businesses such as in the mobile sector (e.g. revenue sharing between app developers and mobile operating system providers). Figure 3.11 accentuates that only less than 20% of software firms share significant parts of their revenues with other stakeholders. Notably, the number of firms sharing more than 40% with partners is pretty low.

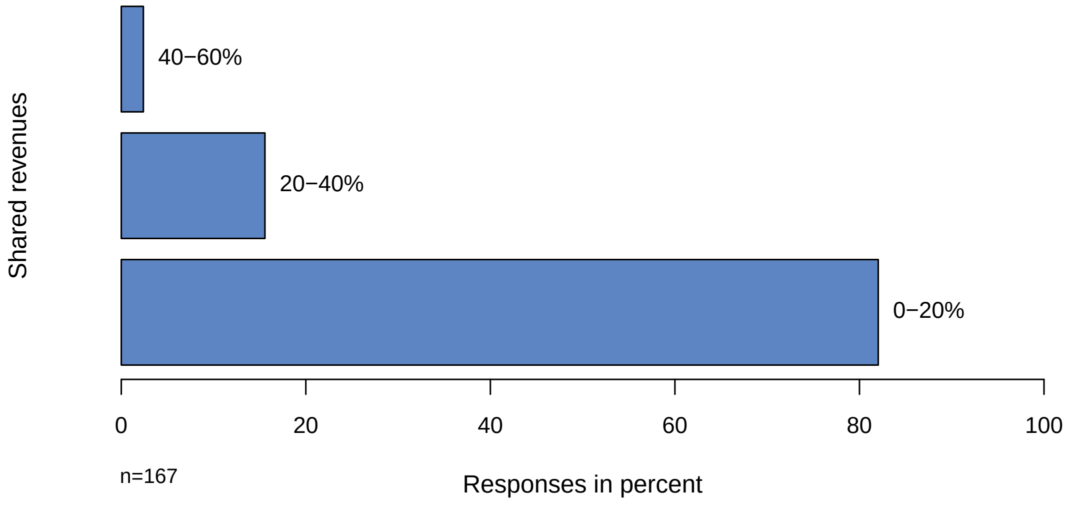


Figure 3.11.: Revenue Distribution Model.

3.3 Upstream: The Solution Composition

Next, we focus the solution composition of software products and services. Again multiple aspects need to be considered and we focus on five key components:

3.3.1 Software Stack Layer

When investigating the properties of software solutions, one fundamental aspect is the type of software that is offered. A very common classification schema is based on a software stack concept (Gao and Iyer, 2006). Software solutions range from low level infrastructure software up to the highest level of application software. The rationale behind this concept is that upper layers build upon the lower level layers. In other words, to run application software, lower level software (e.g. operating system) is required. In Figure 3.12 we investigate the distribution of software solutions according to the mentioned software stack concept. The results show that 40% of the respondents focus on pure application software. In contrast, only 16% classify their solutions as infrastructure software. While the majority of offerings is hence application-centric, several solutions are somewhere in between application and infrastructure software and span several layers according to the software stack concept.

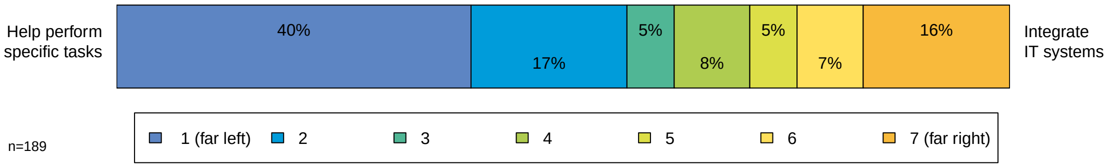


Figure 3.12.: Solution type of the offered product or service.

The high rate of application software firms compared to infrastructure firms makes sense as typically various application software solutions are offered on top of one infrastructure platform. The number of infrastructure solutions hence tends to be lower than the number of application software solutions. Nevertheless, the potential of infrastructure solutions is often enormous as they can attract multiple providers of on-top applications.

3.3.2 Platform

The number of platforms has gradually increased in recent years. Particularly, mobile and cloud computing are well-known trends that shift the traditional platform focus of software solutions. While in the past software was mainly installed on servers and desktop/laptop computers, other platforms have become more and more popular. While these trends are highly cited in press, it is very interesting to analyze the actual market penetration of these platforms.

Figure 3.13 shows the distribution among the different platforms. In our sample, game console platforms are hardly used for software deployment. Also software based on social media platforms and embedded systems is to most companies only of minor importance. The well-known trends of mobile and cloud computing achieve medium average importance values. The traditional platforms, servers and desktop/laptop computers, are still considered to be the most important platforms achieving both the highest valuest by far.

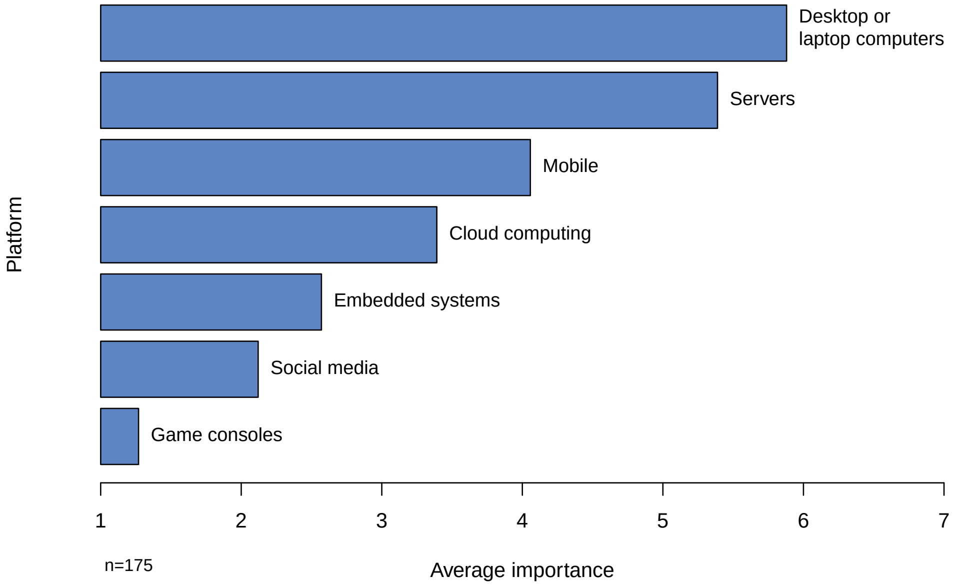


Figure 3.13.: Platforms for which the product or service is offered.

All in all, these results provide interesting insights. As of today, the dominating platforms are still the traditional platforms (i.e. servers and desktop/laptop computers). Notably, this finding is in line with the results of last year's German Software Industry Survey. Interestingly, the high expectations of software firms raised last year (in 2013 emerging platforms (i.e. mobile and cloud computing) may catch up with the traditional platforms) have not been fullfilled to date. Accordingly, the question is by when the expectations will be realized and which speed of growth will be realized in cloud computing and mobile. As of now, it seems that the current kings of the hill (servers and desktop/laptop computers) still defend their important positions.

3.3.3 License Model

The license model differentiates the firms' solutions from a legal perspective. The results to this question are shown in Figure 3.14. In our sample, the most important license type by far is a proprietary license, i.e. firms sell usage rights of their software without disclosing the software code. The three other license types achieve pretty similar values. They are highly important to a limited number of software firms (ca. 20%), but only of limited importance to the majority of firms.

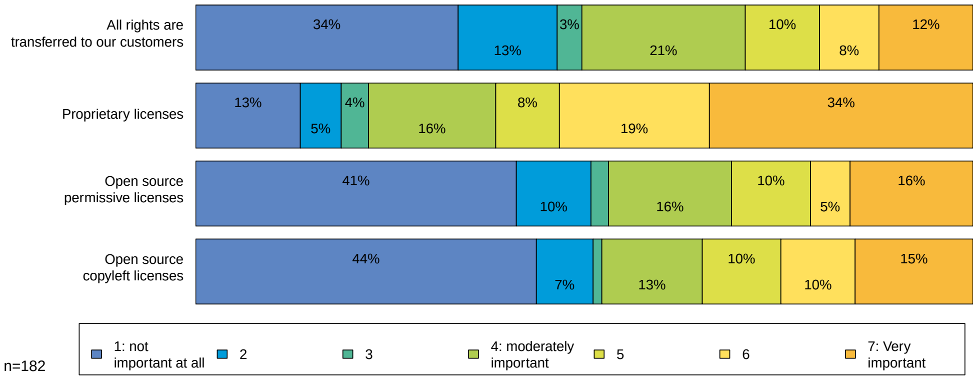


Figure 3.14.: License model of the offered product or service.

3.3.4 Degree of Standardization

In general, standard software providers can be differentiated from customer specific vendors. In our study, we investigate a solution's degree of standardization by asking if the solution in tailor-made for each customer (see Figure 3.15).

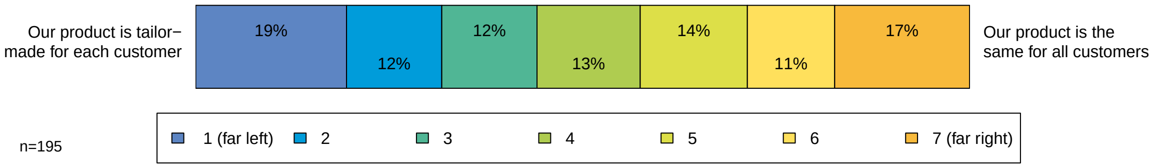


Figure 3.15.: Standardization of the offered product or service.

Interestingly, the distribution of responses is pretty evenly distributed. So, there are firms offering highly standardized products, others are highly customer-specific, and there are offerings in between. Interestingly, the results have somewhat changed compared to last year's Software Industry Survey where the degree of tailor-made solutions was significantly higher. Thus, more firms seem to strive standardizing their offerings. The rationale behind this strategy might be economies of scale effects. Nevertheless, still many firms develop customer-specific software or position themselves in the middle between a pure standard software provider and a highly customer specific developer.

Reflecting these results in the context of the platform findings, it turns out that cloud computing faces an important challenge. As many solutions are still customer-specific, cloud computing providers need to include modification and enhancement functionality into their offerings. Cloud computing providers must allow and even support the technical integration of customer specific third party systems. Having said this, the expected rise of cloud computing will depend on its ability to cope with such kinds of challenges.

3.3.5 Key Cost Driver

Last but not least, we analyze the main key cost drivers of software firms. Figure 3.16 depicts three main cost groups. Product functionality, design, quality, and security is by far the main cost driver contributing about 40-60% to total operating costs. Technical infrastructure and equipment as well as product marketing and sales achieve both only average value of about 20%. Particularly, the results of the produt marketing and sales costs are interesting as particularly SaaS firms are often expected to have high marketing and sales costs. Nevertheless, our findings accentuate the importance of development costs, which imply salaries for product developers and associated job functions.

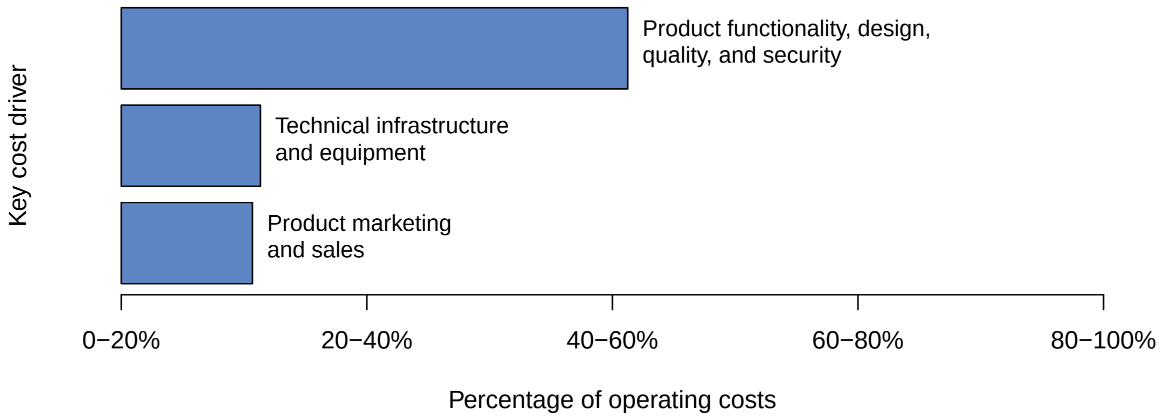


Figure 3.16.: Key Cost Driver.

3.4 Downstream: The Go-To-Market

Next, we focus the go-to-market of software products and services. Again we investigate the business model characteristics of five key components:

3.4.1 Localization

One of the main dimensions of a business model is the target geographic market representing the globalization of software firms. Figure 3.17 depicts the importance of major geographic sales regions. Not surprisingly, Germany is served by almost all firms. Second place goes to the rest of Europe and Russia. Thus, geographic proximity seems to matter in internationalization strategies. Among the three remaining geographic areas, North, South and Central America is served by more software firms than Asia and South Pacific as well as Middle East and Africa. Last year's German Software Industry Survey as well as the results of the Finnish Software Industry Survey show similar results. Apart from their home market, most of the Finnish software firms also achieve revenues in European countries.

3.4.2 Target Customer and User

In this study, we differentiate the customer type and size. We distinguish consumers from business customers as well as small and midsized firms from large firms. The results are shown in 3.18 and emphasize that most software firms target business users. In contrast, private individuals are only of minor importance to most software firms in our sample. With respect to customer size the results show that both types of customer segments are almost equally important with a small plus for large organizations. Thus, most software firms sell to both segments.

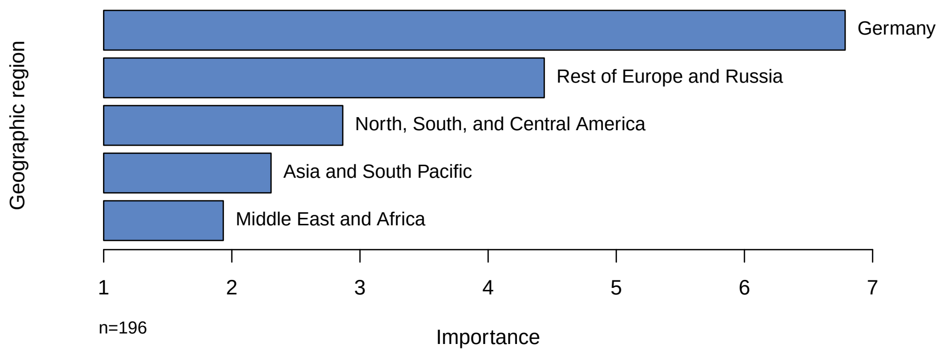


Figure 3.17.: Geographic region where revenues is generated.

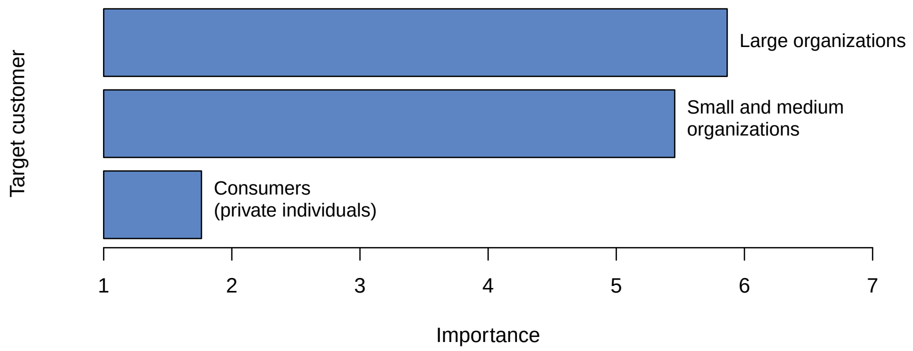


Figure 3.18.: Target customer groups to which the product or service is sold.

3.4.3 Target Industry

This section deepens the analysis of the business customer segment. Figure 3.19 depicts the distribution of target industries. Based on the results of our last year's survey, this year we limit the analysis to the seven most important industries. In line with last year's results, the two dominating industries are still manufacturing as well as information and communication. While the information and communication industry implies that many software solutions are provided to firms within the broader context of the software sector, the rationale for manufacturing may refer to the German industry structure. Manufacturing comprises industries such as the Automobile sector and is hence one of the dominating industries in Germany. Differences among the other five industries are rather small; all industries achieve medium average values showing their relative importance. The medium levels of importance also indicate a certain degree of diversity. In other words, software is offered to various heterogeneous industries.

3.4.4 Channel

After reviewing the different target market properties, one further crucial component of a business model is the appropriate sales channel. According to Figure 3.20, most of the respondents' firms rely on events and sales agents as sales channel. Referring to the fact that most of the firms in our sample sell business software, the dominance of these two channel types is not surprising. Telesales also allows high level of importance allowing a direct customer contact at lower cost than personal sales agents. With increasing number of customers, these channel type usually becomes more important. In contrast, retail stores do hardly matter. Again this fact may refer to the low representation of consumer software firms in our sample. Interestingly, the usage of online shops as the primary channel is not that important. Reflecting the (as of today yet) low number of mobile and cloud computing offerings, this result can be justified. Nevertheless, assuming high growth rates in these platforms, distribution channels are expected to be affected. Particularly mobile solutions are usually mainly sold through online shops.

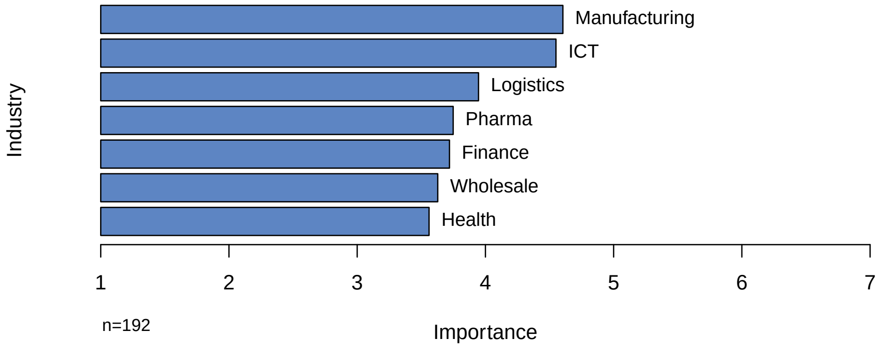


Figure 3.19.: Target industries for which the product or service is offered.

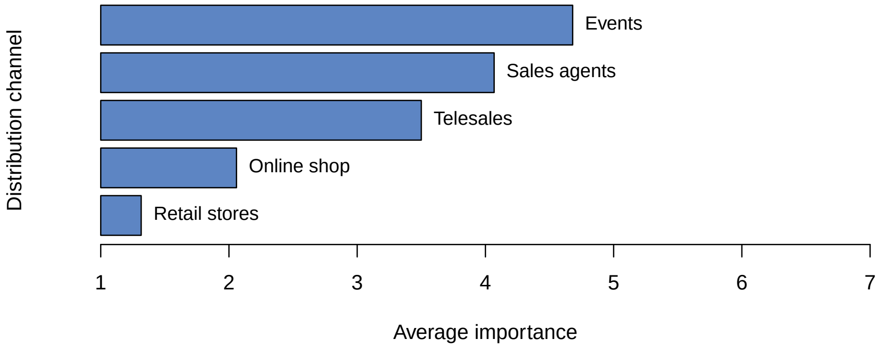


Figure 3.20.: Channels which are used to distribute a firm's product or service.

3.5 Usage: The Solution Lifecycle Management

Finally, the lifecycle management of software products and services is examined by analyzing five key business model components:

3.5.1 Implementation Effort and Operating Model

Implementation and operation of software solutions become relevant in later phases of a products' lifecycle, when it is near to its actual usage or is in use already. In this year's survey we asked the respondents how their firms operate their product or service and how high they would rate the implementation effort. The results are shown in Figure 3.21.

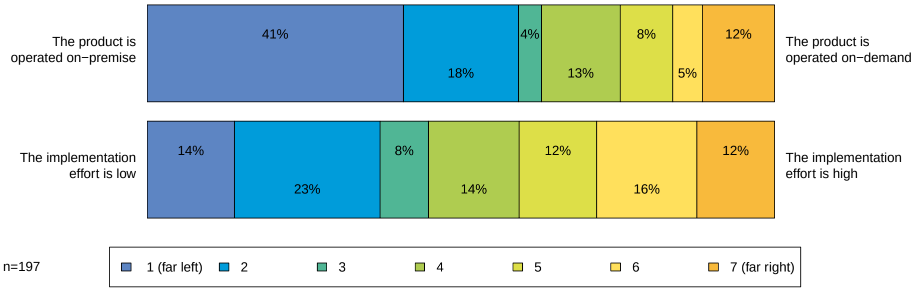


Figure 3.21.: Operation and implementation effort of the offered product or service.

Whereas solutions, which are operated on-premise are best comparable with the traditional way of installing and using a software, on-demand solutions are provided over the internet. Thus, we consider on-demand solutions as substitute term for SaaS and cloud computing offerings. Given the hype around cloud computing, we are interested in the actual share of on-demand solutions. The results show that only a minority can be classified as pure on-demand solutions, whereas on-premise accounts for most responses. The difference is still considerable and in line with the findings of last year's Software Industry Survey.

Many solutions require implementation effort before they can be actually used. Our results indicate that the implementation effort varies, but is pretty evenly distributed. We thus take a closer look in order to determine which factors influence the implementation effort of a solution. For that, Figure 3.22 shows the implementation effort of on-demand solutions vis-a-vis on-premise solutions. We would expect that on-demand solutions should require less implementation effort as they are often advertised as more standardized and easier to use. While the results do show some difference with regard to the required implementation effort, it is not that large as often expected. The often stated assumption that on-demand solutions require much less implementation efforts seems hence to be questionable.

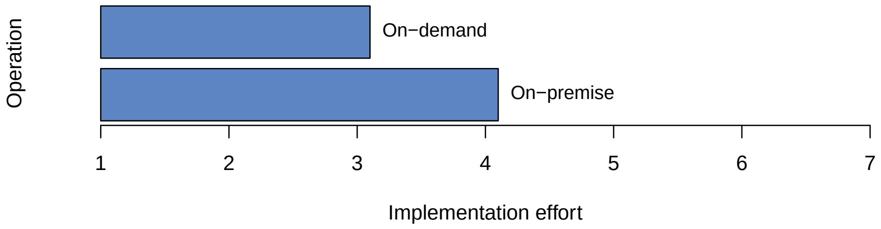


Figure 3.22.: Implementation effort of on-premise and on-demand solutions.

3.5.2 Maintenance Model

In terms of maintenance we examine the strategy of software firms by measuring their release frequency. The release frequency as depicted in Figure 3.23 shows that most firms offer new releases once a year or even less often. Less than 20% follow an agile release delivery principle and publish new releases weekly or even more often. Among the other three categories, one to three release cycles are most common. The results show that the release cycles are highly spread among the sample firms.

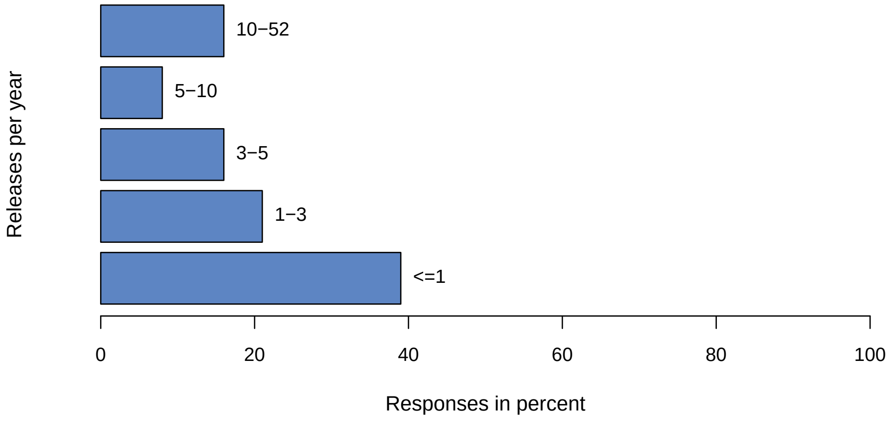


Figure 3.23.: Release frequency of the offered product or service.

3.5.3 Support Model

Next, we investigate the support offerings of our sample firms by examining the standardization of support contracts. The results are shown in Figure 3.24.

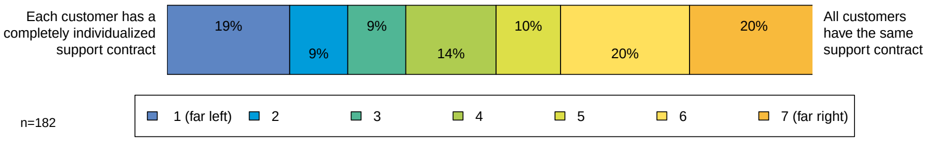


Figure 3.24.: Characteristics of the support offered.

Firms with standardized support contracts will benefit from economies of scale, whereas individualized contracts allow for more differentiated customer treatments. We can see that both extremes are about equal and the distribution in total is pretty even. There is just a very slight tendency toward standardized support contracts. It hence appears that the diversity of support contract models is rather high and no single conclusion can be drawn for the industry as a whole.

3.5.4 Replacement Strategy

Finally, we analyze the number of releases that are available on the market at a time (see Figure 3.25). Most of the software firms offer only one release at a time. In other words, they manage to keep all customers on the same release level being a beneficial strategy in terms of operating and maintenance efforts. Nevertheless, almost 60% do not manage to keep all customers on the same release level. Some firms even have customers running on more than 5 releases at a time.

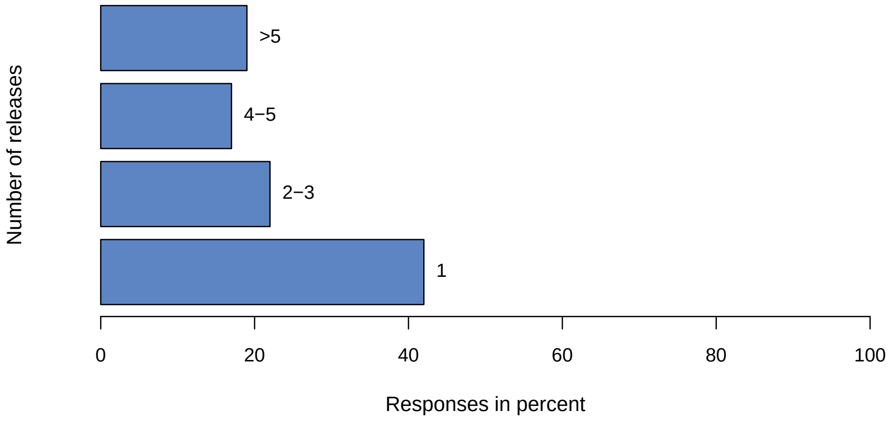


Figure 3.25.: Parallel releases in the marketplace of a product or service.

4 Strategic Groups: Business Model Classes and Their Performance

Whereas each firm can have an individual business model, it is often possible to reduce the overall number of individual firms to few classes[1] in order to reduce complexity. The result is a *classification framework* for firms, where each class summarizes the main characteristics of the firms belonging to this class. E.g., it is common to differentiate individual and standard software product firms and associate them with certain characteristics such as reusability and adaptability.

Classifications have multiple applications in research and practice. In general, they can be used for industry analysis as well as individual firm analysis. In industry analysis, an understanding of prevalent business model classes and most distinguishing characteristics can be gained. This information can be used to position one's firm accordingly or to identify relevant competitors in an industry. In firm analysis, one can identify distinctive characteristics between firms and explain intra-industry differences in performance.

In our survey, each respondent could classify her firm according to a given classification scheme. However, such pre-selected classification schemes have several limitations as they limit the answering options. They further do not necessarily describe the business models of the firms. Our main objective was to derive an analytical classification which is only based on empirical business model data and mathematical methods, thus removing subjective perceptions (e.g. from researchers and practitioners). As we were particularly interested in core software firms, we used firms who identified themselves as individual development or standard software firms for the analyses presented in this chapter.

In order to identify business model classes in the German software industry, we performed a *strategic group analysis*. The basic idea is to view an industry "as composed of . . . groups of firms, where each group consists of firms following similar strategies in terms of the key decision variable" (Porter, 1979, p. 215). We use the terms *business model class* and *strategic group* interchangeably, but prefer the latter term in this chapter and define it as follows:

> A **strategic group** is a group of firms following similar business models.

In what follows, we first describe the three main strategic groups in the German software industry which we found during the course of our analyses. We then compare the performance of the strategic groups.

4.1 Strategic Group Characteristics

In order to identify strategic groups, we grouped firms with similar business model characteristics[2]. By reviewing the main characteristics of each group, we could label them with meaningful names as presented in Figure 4.1. Table 4.2 summarizes the characteristics of each group.

		Operation mode	
		On-premise	On-demand
Standardization level	Standard	1 On-premise standard software	2 On-demand standard software
	Individual	3 Individual software	

Table 4.1.: Strategic groups organized along two dimensions.

Figure 4.1 shows that two groups provide standard software and one group individual software. Standard software firms can be further divided in on-premise and on-demand software. Accordingly, we termed the strategic groups as (1) on-premise standard software, (2) on-demand standard software, and (3) individual software. While these labels are common to describe software firms, we note that in our case these labels represent a broad collection of business model

[1] Other common terms include: archetypes, clusters, groups, types, or design schemes.

[2] The mathematical method used here was a cluster analysis. In the first step, a hierarchical agglomerative clustering (with Ward method) was used to find the optimal number of clusters. Then, we used k-medoids to form the final clusters given the optimal number of groups.

Characteristic	Strategic group			Number	Kruskal-wallis test	
	1	2	3	of values	across groups (p-value)	
Value Proposition: Price	3.51	3.86	4.60	77	0.0067	**
Revenue Distribution Model: High	1.05	1.71	1.08	68	0.0000	***
Value Chain: Research	0.35	0.08	0.37	77	0.0437	*
Value Chain: Maintenance	0.65	0.71	0.11	77	0.0000	***
Value Chain: Marketing	0.04	0.12	0.37	77	0.0054	**
Value Chain: Implementation	0.65	0.25	0.48	77	0.0174	*
Value Chain: Support	0.19	0.54	0.33	77	0.0358	*
Revenue Source: Third Parties	1.42	1.71	3.40	75	0.0001	***
Payment Flow Structure: Recurrin	2.46	4.29	4.93	77	0.0002	***
Degree of Standardization: High	4.46	5.33	2.35	76	0.0000	***
Implementation Effort: High	4.54	2.50	4.04	76	0.0014	**
Operating Model: On-demand	1.65	2.88	3.84	75	0.0021	**
Support Model: Standard Support	5.12	4.92	3.44	75	0.0211	*
Platform: Mobile	3.42	3.58	5.07	77	0.0049	**
Platform: Cloud Computing	2.46	3.29	4.56	77	0.0042	**
Target Industry: Single Industry	3.54	3.04	1.60	76	0.0017	**
Channel: Online Shops	1.04	2.74	2.26	76	0.0008	***

***p<0.001; **p<0.01; *p<0.05; †p<0.1

Table 4.2.: Strategic group characteristics (all values in range [1-7]). Only group characteristics with significant differences across groups are shown (at level p<0.05).

characteristics. These three groups further only identify the most prevalent business models in the software industry. They do not exclude the existence of other, less common business models.

Based on the statistics in Table 4.2 we characterize the strategic groups (i.e., the business models of the firms in these groups) as follows:

1. **On-premise standard software** firms use what may be regarded as the most classical type of business model: They offer standardized software products which are operated on-premise. Keeping in mind that most firms in our sample are B2B firms, several characteristics fit the description of business software: the product requires high implementation effort and it ranks comparably high with regard to one specific target industry. Online shops are not an important distribution channel, as these products are too complex to be distributed without individual sales contact. We further note that the pricing model ranks high with regard to revenue generation from end-users and single-payments (as opposed to recurring payments). The main value chain activities include research, maintenance, and implementation. Surprisingly, support ranks low when compared to other groups. However, this just shows that other activities are regarded as more value-creating.

2. **On-demand standard software** firms provide there product as a service by making it available to their customers via the internet. This group is made up of what is generally referred to as cloud computing and software-as-a-service. As these firms often operate their products themselves, operations is seen as a major value-creating activity when compared to other groups. Also the product often operated as on-demand rather than on-premise. This allows the firms to make use of usage-based pricing with recurrent payments. As the product is often provided as a website, advertisements can be included or valuable data collected, such that revenues from third parties can be generated. Interestingly, the solutions of these firms have low standardization and low verticalization.

3. **Individual software** firms offer solutions with a low degree of standardization. Their main value-creating activities include research and marketing. Implementation and support also rank high, whereas maintenance ranks low. The importance of implementation and support also shows in the required implementation effort of the product as well as in the comparably individualized support contracts. Overall, the value chain activities show that firms in this group pursue new projects which require novel solutios. By implementing the solutions or offering support, these firms generate revenues, whereas revenues from maintenance contracts are presumably of lower importance. As these activities are service-intensive they do not scale well, which explains why these firms are particularly interested in keeping their costs low. Interestingly, individual software firms rank highest with regard to revenues from third parties (as opposed to end-customers). Our interpretation is that these firms more often create software for customers who are not the end-customers of the product, e.g. internet portals. This is somewhat supported by other group characteristics such as on-demand operation of the product (e.g. software-as-a-service) as well

as target platforms (mobile and cloud computing). Both characteristics further suggest that these novel trends require expertise and custom solutions, which is offered by individual software firms. As verticalization is low in number and rank, this shows that individual software firms do not specialize in particular industries. Finally, looking at the usage of online channels, we can see that some firms in this group make use of online channels, which appears difficult given the nature of individual software. However, the average importance of 2.26 shows that online channels are more an exception than a rule.

Though the resulting groups are not very surprising, the results confirm common terms to classify software firms (e.g., software-as-a-service firms can be found in the third group). While there are certainly firms that do not necessarily fit into any of the groups, our empirical findings show the most prevalent strategic groups and the characteristics of the groups help to describe the business models of firms in these groups.

4.2 Strategic Group Performance

It is of particular interest to practitioners which strategic groups are most favorable, such that firms can position themselves accordingly. We estimate *performance* by looking at returns and risk of all firms within a group. We further calculate a risk-adjusted return figure:

1. **Return** denotes common business outcome variables such as profits and revenues. We measured return by asking the participants how they estimate their product/service returns toward their main competitors.

2. **Risk** denotes common business outcome variables such as profits and revenues. It is defined as the "unpredictability of business outcome variables" (Bromiley et al., 2001, p. 261). We measured risk by asking the participants how they estimate the volatility of their product/service returns toward their main competitors. Return volatility is commonly seen as an appropriate measure of risk as it reflects the unpredictability of major business outcome variables.

3. **Risk-adjusted return** is calculated by dividing return by risk, such that high risk reduces risk-adjusted performance and vice-versa.

The results are shown in Table 4.3. First of all, we note that all performance measure are significant, showing that there are performance differences across groups. The number of responses is much lower for the risk figure though, as participants probably found it more difficult to estimate these measures.

Characteristic	Strategic group			Number	Kruskal-wallis test	
	1	2	3	of values	across groups (p-value)	
Return	4.42	3.39	4.00	72	0.0275	*
Risk	3.54	4.17	3.52	61	0.0666	†
Risk-adjusted return	1.29	0.93	1.31	61	0.0175	*

***p<0.001; **p<0.01; *p<0.05; †p<0.1

Table 4.3.: Strategic group performance.

Interestingly, we find that the on-demand software group ranks lowest in all performance measures. This is somewhat surprising because this business model is most novel and some well-known firms are transforming from traditional business models to on-demand software offerings. However, this might also explain the relatively low performance, as on-demand software requires high investments in an uncertain environment (e.g., changing competitive environment and technology), thus offering potential high return in the future with the risk of returns not becoming reality. An analysis of changes in performance could shed more light on how on-demand software develops in comparison to traditional business models, though we do not possess such data at this moment.

Looking at the traditional business models in strategic group one and three, we find no significant differences. Even though the return measure is higher for on-premise software firms, this difference in performance disappears after adjustment for risk[3]. Initially, we would have expected higher returns for standard software firms as opposed to individual software, because standard software has very low reproduction costs (Buxmann et al., 2012). However, individual firms counter higher reproduction costs by paying higher attention to their costs. We can further think that the average standard software firm does not scale sufficiently to make up for the high initial development costs, such that on average both groups perform about the same.

[3] This might seem mathematically unlikely as the risk figure is nearly the same for both groups. However, it appears that some high-return firms in the on-premise software group did not provide an estimate of risk. Consequently, risk-adjusted return could not be calculated for these firms.

5 Software Ecosystems

The Software Industry Survey 2013 aimed not only at software producing entities but also at the wider population of firms around them: the “software ecosystem”. (Jansen et al., 2013, p.29) define a software ecosystem as follows:

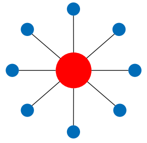
> **Software ecosystem:** a set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. These relationships are frequently underpinned by a common technological platform or market and operate through the exchange of information, resources and artifacts.

The survey looked into several aspects of these ecosystems:

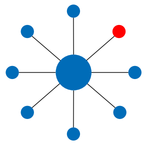
- Role choice: firms in software ecosystems can take on different roles. Our first results suggest typical choice patterns depending on the type of software produced. (Section 5.1)
- Partnerships: hub-spoke relationships are a typical structure found in software ecosystems. A large software vendor (hub) offers partnerships to related firms (spokes). The survey reveals the most prominent hubs and shows how the spokes’ offerings differ from the hub’s offerings. (Section 5.2)
- Partnering motives: spoke firms are most interested in their respective hub’s technology and reputation for entering a partnership. Overall, they largely see their partnerhip as a success and plan to continue. (Section 5.3)
- Dangers of partnering: in general, spoke firms seem not too worried to be cannibalized by their respective hub(s). They mainly rely on their expertise and strong customer relationships to secure their business. (Section 5.4)

5.1 Software Ecosystem Roles

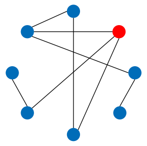
Many authors in the area of business ecosystems use the picture of biological ecosystems when characterizing firm networks: as species in nature, firms in business ecosystems have to find their place to survive. Iansiti and Levien (2004), for instance, stay close to the picture when categorizing firms as keystones, dominators, or niche players and drawing parallels with ecology. For the purpose of this survey and its focus on partnerships, participating firms were asked to classify themselves into one of the following roles:



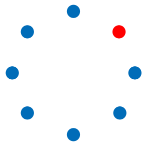
Hub: A hub is the central and dominating firm in a network of partners. It provides the central product or platform based on which other companies (the spokes) do business.
Examples: SAP ERP, Apple iOS, Valve Steam



Spoke: A spoke relies on the product or platform of a hub to perform its business; the relationship is often formalized by joining a partner-, developer-, or reseller-program.
Examples: SAP consultant, app developer, Microsoft distributor, game developer



Network of equals: In this type of network, there is no dominant player but firms collaborate closely and over longer time periods.
Examples: decentralized open source projects such as Linux, Eclipse, KDE



Independent: Independent software firms only develop weak ties to partners if needed, typically relationships are temporary.

The split of participating firms into these roles is shown in Figure 5.1. As was to be expected, hubs account for the smallest share (18%), while most firms classify themselves as being independent (31%). The analysis becomes more interesting when the firm type (Figure 2.6) is considered in addition. Figure 5.2 displays the role distribution per firm category.

Manufacturers of standard software can be found occupying all roles in the software ecosystem, most of them operate independently and thus keep all activities under their own control. Firms in a “network of equals” role jointly develop standard software with partners, but without a dominant player. Most notably, however, standard software features

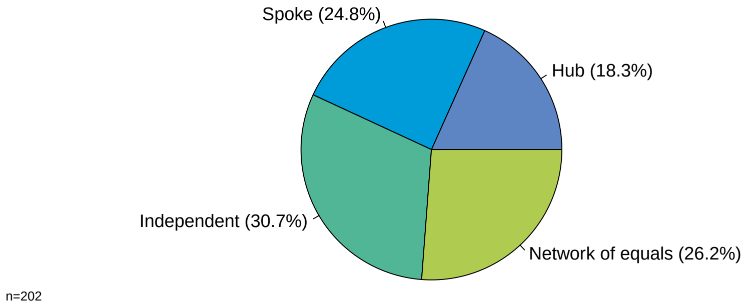


Figure 5.1.: Split of firms into software ecosystem roles

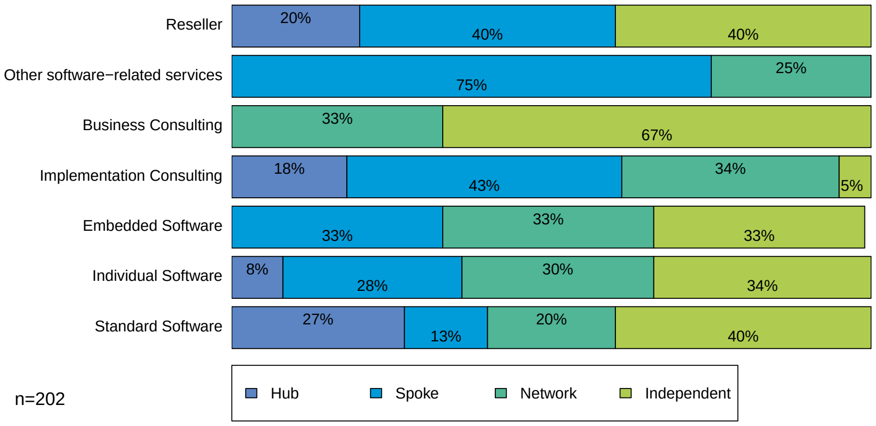


Figure 5.2.: Ecosystem roles by firm type

the highest share of "hub" in all categories and represents 65% of the hubs participating in the survey. Large standard software manufacturers traditionally rely on partners to implement, customize, enhance and sell their products, thus constituting a typical hub.

Individual software, in contrast, is rarely developed by firms in a hub role. Manufacturers of individual software seem to prefer networked, spoke, or independent setups which might allow for the flexibility and agility needed for individual software development.

Mirroring the high share of standard software firms in hubs, spoke is the dominating role chosen by firms in embedded software (building on a proprietary hub technology platform), implementation consulting (where firms often develop expert and niche knowledge in a limited range of software products), other services, and resellers. The 20% share of "hub" in resellers is an indication of multi-tier distribution, which was not asked for separately in the survey.

Business consultants stand out in their high share of "independent" setups (67%), while none of the respondents is in a spoke role. Providing manufacturer-independent advice to clients seems a characteristic task of this type of consulting.

5.2 Hub-Spoke Partnerships

A typical partnership pattern in software ecosystems occurs when a large vendor acts as a hub, attracting different types of partners filling niches around its products. These spokes can produce complementary software assets (such as, for example, smartphone apps), but many also provide related services or act as resellers. We asked spokes to name the hub(s) most important to their business. Figure 5.3 depicts the ranking of hub mentionings in our sample.

Microsoft is clearly in the lead, which is no surprise given its strong market position and sophisticated partner program. More suprising is the second position, which comprises "other" hubs not offered in the list of choices. The free-text responses for this category do not identify further strong hubs but show a broad set of software manufacturers acting as hubs in niche markets. The existence of hub-spoke relationships aside of the main market is an important result of this study. The further places are held by enterprise software manufacturers (Oracle, SAP), where consultants represent

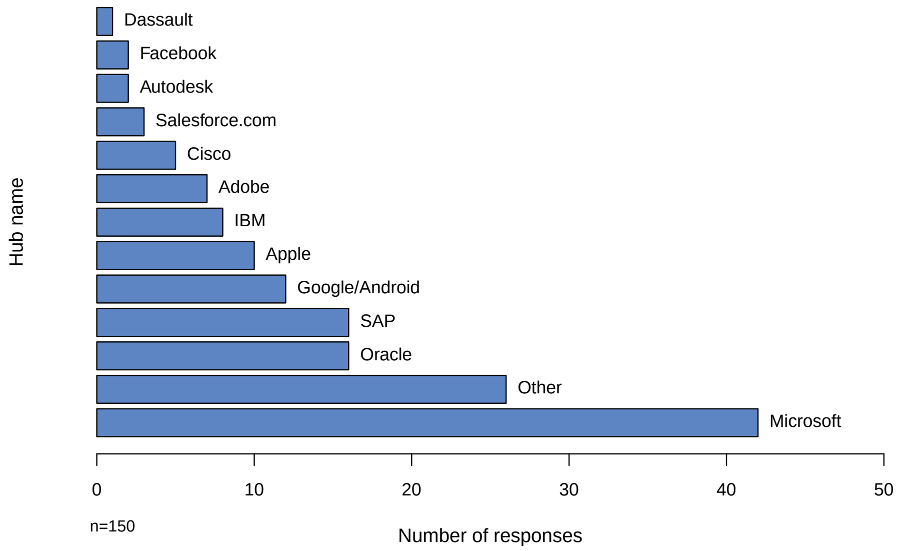


Figure 5.3.: Popularity of hubs (number of mentions)

a strong spoke category. Smartphone apps (indicated by the hubs Google and Apple) are an important hub-spoke-type market as well, which has grown rapidly over the past few years.

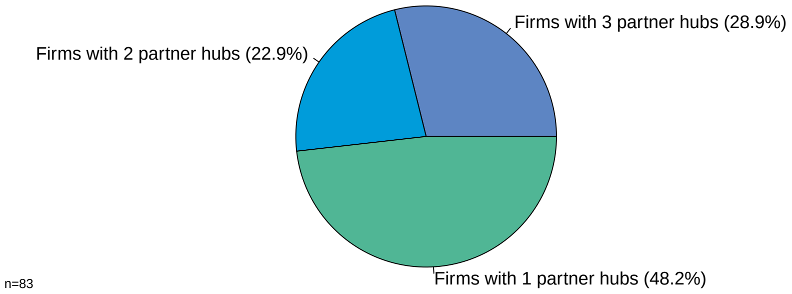


Figure 5.4.: Number of hub partnerships per spoke

Overall, about half of the spokes indicated to cooperate with two or more hubs (Figure 5.4), which might hint at a multi-hub strategy. Roughly the other half of spokes partners with one hub exclusively, placing all of their bets on one vendor. We will come back to this point in Section 5.4. Figure 5.5 illustrates how much the offerings of the hub and the spoke overlap, according to the spokes. It is interesting to see that the two extremes are both almost equally distributed. A spoke either offers the same or similar products as the hub, potentially differentiating through niche market knowledge, or it offers complementary products which extend and complement the hub's offerings.

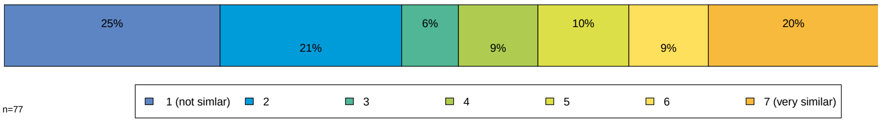


Figure 5.5.: Complementarity of hub and spoke offerings

5.3 Reasons for Partnering

Given that most software firms are independent (see Section 5.1), it is a valid question to ask why other firms choose to become, to a certain degree, dependent on a hub's products instead of keeping value creation under their own control.

We asked spokes for the importance of five dimensions of factors for their decision to become a hub's partner. Figure 5.6 illustrates that it is mainly the hub's technology base and the hub's reputation in the market which are important to spokes. The other three factors (direct support such as marketing grants, market access to hub customers and the hub's innovation capability) are also important but to a lesser extent.

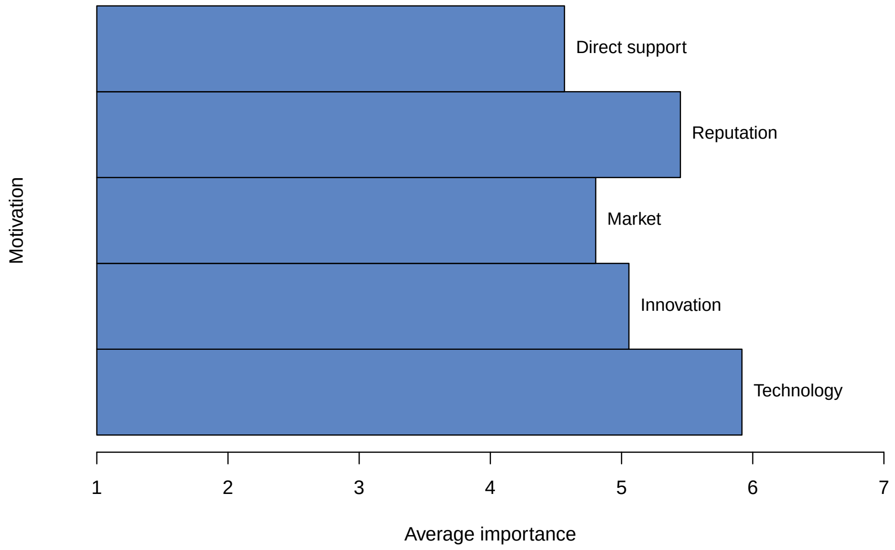


Figure 5.6.: Reasons for spokes to become a hub partner

5.4 Dangers of Partnering

It is a common assumption in literature that uneven partnerships, such as hub-spoke relationships, entail risks for the smaller partner (e.g., Taylor, 2005). The hub could, for example, use its power to take over know-how, personnel, or even clients from the spoke. The respondents in our study, by the majority, do not see these risks and most of them agree that their and the hub's interests are aligned (see Figure 5.7). It becomes very clear, however, that spokes see their strengths in niche and expert knowledge and, even more, in strong customer relationships. This focus brings them into a position that is hard for the hub to occupy itself, which might explain the alignment of interests.

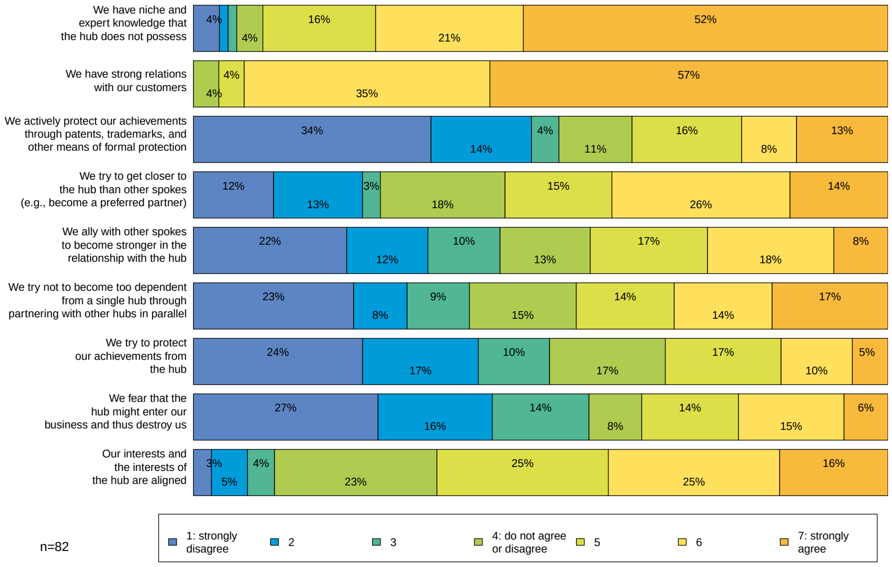


Figure 5.7.: Risks of partnering and counter measures

Formal means of protection, such as patents or trademarks, are rarely used. Other strategies, such as allying with other spokes or partnering with multiple hubs at the same time, are used but only by a minority of spokes.

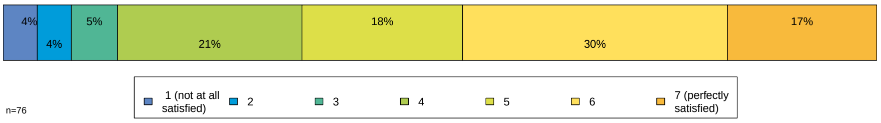


Figure 5.8.: Satisfaction of spokes with their hub partnership

Concluding this section on software ecosystems, it can be noted that the benefits of partnering seem to prevail. Despite the risks mentioned, the majority of responding spokes indicated their satisfaction with the partnership and intends to continue (see Figure 5.8).

A Appendix: Research Methods

Developing the Contact List

The primary data for this study was collected from the German software industry using a web questionnaire. We aimed to cover the entire software industry, including firms that do software business but are not necessarily classified as software firms in the official industry classification. We used the Orbis database, which we queried with the NACE Rev. 2 industry codes 62, 63 and 582, as the main source of data to compile the sampling frame and to get up-to-date contact information and financial data.

After completing these steps, the long list of firms included more than 33,000 firms. The entire list was screened to exclude firms to which we felt that it was not appropriate to send the survey. After eliminating firms that were no longer active, removing non-software firms, and combining firms that were actually just one firm (e.g. corporations with subsidiaries or holding firms), a total of 32,611 firms were included in the contact database. During the course of the project, several hundred firms were removed since they reported not being active or not being in the scope of the survey. Missing e-mail addresses were collected from firm websites. We mostly contacted the firms through their general contact addresses.

Data Collection Process

The survey was implemented following a modified version of the tailored survey design method (Dillman et al., 2009). The data collection process began by sending out the main survey package to all firms on 2nd April 2013. The roll-out mail contained information about the survey and instructions on how to participate in the survey. The delivery status of the emails was recorded, and a second batch of emails were sent a few weeks later. In the second emailing, all non-functioning emails in the first round (i.e., emails that bounced from the receiving mail servers) were removed or substituted with new untried email addresses, if available.

During the emailing of the survey, we were informed of several firms who had moved, whose contact person was no longer working for the firm, or who were not operating as an independent firm anymore. All undelivered emails were analyzed, and where possible, the contact was repeated using new contact information of the firm.

Several approaches were taken to convince the informant of the importance of the survey. Many organizations closely linked to the software industry were asked to endorse the survey. The survey was conducted on behalf of the Software-Cluster, which is funded by the German Federal Ministry of Education and Research. In addition, we promised to provide firm-specific reports of the responses as a further incentive to respond.

The total number of responses was 427. Besides 219 complete responses, we obtained responses in which the response was only partial; that is, responded to some questions but not proceeded to the end of the survey form. These partial responses were obtained from 208 firms.

Though investigating the German software industry, our sample may include some firms from abroad. The rationale for this effect is that some firms are listed as German software firms though being a subsidiary of an international parent organization. When investigating the structure of our sample we identified a few firms that may fall into this category. Nevertheless, as this only occurs very rarely, we see our sample still as a good proxy for the German software industry.

Development of the Survey Instrument

The survey was developed in cooperation with our cooperation partners from Aalto University, who have been conducting a survey in Finland since 2002. Whenever possible, we adapted questions from their previous survey or from scientific publications. Whenever this was not possible, we conducted our own lists of possible measurements and selected appropriate measures after discussions in our research group. The survey was conducted in German and English. The original questionnaire was designed in English and then translated to English using an adapted back-translate procedure (Brislin, 1970).

Data Preparation and Analysis

Prior to data analysis, all data form this year was combined into one database. Actual data preparation and analysis was carried out using R statistic software.

In addition to the first elimination of clerical errors and outliers, we performed separate elimination of outliers for each analysis. In most cases, we used five standard deviations as a threshold for determining outlier values (an exception is e.g. firm age, where we could review the responses and verify that the outliers are correct data). This rule was applied iteratively until no more values were eliminated. The amount of data used in each analysis is presented with the results after outliers have been eliminated.

Statistical analyses that were used are explained in text or in the footers of results tables and figures. Since all of these methods are de-facto standards for this kind of report and good descriptions are available elsewhere, the description of these methods is omitted from this report. In this study the effect of non-response was not systematically analyzed due to lack of resources, unless otherwise noted under individual analyses.

Bibliography

Brislin, R. W. (1970). Back-translation for cross-cultural research. *Journal of Cross-Cultural Psychology*, 3(1):185–216.

Bromiley, P., Miller, K. D., and Rau, D. (2001). Risk in strategic management research. In Hitt, M. A., Freeman, R. E., and Harrison, J. S., editors, *The Blackwell Handbook of Strategic Management*, pages 259–288. Blackwell, Malden, USA.

Burkhart, T., Werth, D., Krumeich, J., and Loos, P. (2011). Analyzing the business model concept - a comprehensive classification. In *International Conference on Information Systems*, pages 1–19, Shanghai.

Buxmann, P., Diefenbach, H., and Hess, T. (2012). *The Software Industry*. Springer, Berlin.

Dillman, D. A., Smyth, J. D., and Christian, L. M. (2009). *Internet, Mail, and Mixed-Mode Surveys - The Tailored Design Method*. Wiley, Hoboken, USA, 3rd edition.

Gao, L. and Iyer, B. (2006). Analyzing complementarities using software stacks for software industry acquisitions. *Journal of Management Information Systems*, 23(2):119–147.

Iansiti, M. and Levien, R. (2004). Strategy as ecology. *Harvard Business Review*, 82(3):68–81.

Jansen, S., Brinkkemper, S., and Finkelstein, A. (2013). Business network management as a survival strategy. In Jansen, S., Brinkkemper, S., and Cusumano, M. A., editors, *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*, chapter 2, pages 29–42. Edward Elgar Publishing Ltd., Cheltenham, UK.

Lehmann, S. and Buxmann, P. (2009). Pricing strategies of software vendors. *Business & Information Systems Engineering*, 1(6):452–462.

Porter, M. E. (1979). The structure within industries and companies' performance. *The Review of Economics and Statistics*, 61(2):214–227.

Pussep, A., Schief, M., and Buxmann, P. (2012). Results of the German Software Industry Survey 2012. Technical report, Technische Universität Darmstadt, Darmstadt, Germany.

Schief, M. (2013). *The Impact of Business Model Characteristics on Firm and M&A Performance in the Software Industry*. Dissertation, TU Darmstadt.

Schief, M. and Buxmann, P. (2012). Business Models in the Software Industry. In *45th Hawaiian Conference on System Science*, pages 3328–3337. Ieee.

Schleife, K. and Leimbach, T. (2012). Analyse von Wachstumshemmnissen kleiner und mittlerer Unternehmen am Beispiel der IT-Branche. Technical report, Fraunhofer ISI.

Taylor, A. (2005). An operations perspective on strategic alliance success factors: an exploratory study of alliance managers in the software iindustry. *International Journal of Operations & Production Management*, 25(5):469–490.