Information Systems and e-Business Management (2021) 19:281–311
https://doi.org/10.1007/s10257-020-00499-9

ORIGINAL ARTICLE

An analysis of product strategy in cloud transition considering SaaS customization

Zhongdong Xiao[1] · Wenjun Shu[1] · Abigail Osei Owusu[1]

Received: 8 February 2020 / Revised: 29 September 2020 / Accepted: 14 November 2020 / Published online: 27 January 2021

Abstract

When traditional enterprise software vendors adapt to software as a service (SaaS) practices and evolving to cloud service models, there is a major change occurring in the enterprise: a new hybrid product strategy consists of on-premises software and competitive customized SaaS. For the first time, we build a stylized model to reveal the influence of SaaS customization on the decisions of monopoly software vendor in the transition period. Increasing the customization efficiency of SaaS results in two possible structural regimes in the market. One is single on-premises software dominate the market if SaaS is customized at a low level and the other is hybrid products segment the market if SaaS is moderate-level customized. Surprisingly, software vendors with high customization proficiency should not allow SaaS products to dominate the market. It would benefit more from offering a competitive hybrid product strategy. Therefore, this paper does not recommend traditional software vendors to transform into pure cloud service providers. This key findings remain valid in the extended analysis of other customization technologies. Besides, the extension models show that both configuration and personalization customization technologies outperform the modification customization technology.

Keywords Product strategy · Pricing · SaaS customization · Cloud computing · Market structure

1 Introduction

Traditional software enterprises adopt on-premises applications, which is supported by independent teamwork for operations and maintenance. Software-as-a-service (SaaS) is accessed over the Internet and deployed in the cloud as a hosted service (Jaroucheh et al. 2010). SaaS is occupying more and more software markets.

✉ Wenjun Shu
shuwj3034@stu.xjtu.edu.cn

[1] The Key Lab of the Ministry of Education for Process Control & Efficiency Engineering, School of Management, Xi'an Jiaotong University, Xi'an 710049, China

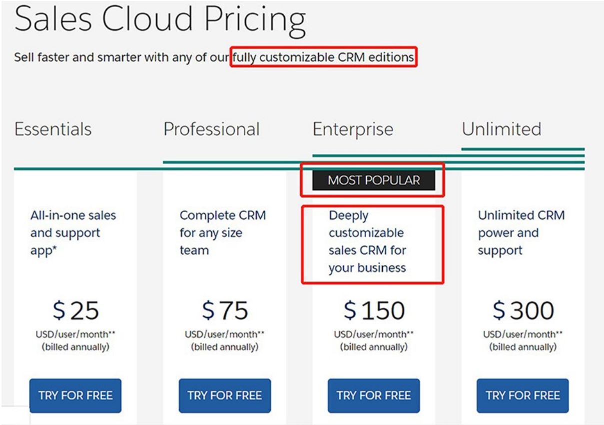

Fig. 1 Salesforce product pricing (see more at: https://www.salesforce.com/cn/products/sales-cloud/pricing/)

Salesforce generated \$13.28 billion in fiscal 2019, which counts 47% of SAP's total revenue.[1] The rapid development of Salesforce reflects the popularity of SaaS business model, which leads to a transition of traditional software vendors. Oracle is jockeying to regain momentum by hiring roughly 2,000 employees to bolster its cloud computing activities, since it has fallen behind Amazon.com and Microsoft in cloud computing (Thomas and Fitch 2019). Besides, SAP's cloud subscriptions and services generated \$4.993 billion in revenue in 2018, surpassing traditional on-premises licensing revenue for the first time[1].

The emergence of software as a service (SaaS) has prompted the traditional software vendors to transit to SaaS providers. Transitional software vendors can provide a single product strategy or a hybrid product strategy. Single product strategy means only on-premises software or SaaS offered, and a hybrid strategy consists of both. Previous research shows that software providers can use SaaS as a competitive strategy in the software market (Ojala 2016). Besides, those vendors have user bases before the transition, with marketing channels and sales teams. Therefore, during the transition, a hybrid product strategy is the choice: offering SaaS products while providing on-premises software by the transitional vendors.

This work focuses on two characteristics of a hybrid product strategy. The first is customization in the SaaS product line, which was considered to be providing only basic functionality due to its multitenant features (Ma and Kauffman 2014; Ma and Seidmann 2015). Compared with on-premises software, SaaS is beneficial to achieve economies of scale, and has the risk of low-security performance. However, by customizing SaaS products, SaaS providers can improve product quality to meet security needs and offer initial cost savings in their practices. Taking CRM software as an example, Salesforce provides popular customized CRM for sale (see Fig. 1),

[1] See more at: https://www.sap.com/docs/download/investors/2018/sap-2018-integrated-report.pdf.

which can save 58% of the cost compared to on-premises CRM applications from Oracle.[2] However, customized SaaS products can hinder the economies of scale of suppliers (Sun et al. 2008; Kong et al. 2010) and lead to high customization investments. Therefore, software vendors in transition need to consider the trade-off between customization needs and economies of scale when making customization decisions.

Secondly, the differences between traditional software products and SaaS products in the hybrid strategy are considered. The most significant difference between the two versions is how they are implemented and who is responsible for their operations and maintenance. SaaS is accessed via the Internet, instead of having it installed and maintained by the IT department. SaaS users only need to subscribe to the application and pay them on demand. Traditional on-premises software requires users to pay a one-time licensing fee for lifetime use, install the IT infrastructure locally and manage the day-to-day operation.

Based on the above characteristics, this work studies explicitly the following issues. (1) Should the transitional software vendor provide customized services for SaaS products? How to determine the level of customization for their SaaS offerings? (2) Under what conditions can transitional enterprises obtain the optimal profit by providing a hybrid product strategy? (3) How does the degree of customization of SaaS products affect software providers' product decisions, pricing strategies, and profits during the transition period? (4) Will the market eventually be dominated by SaaS products?

This paper studies the optimal product strategy decision of a monopoly software provider during the transition period. It focuses on the competitive dynamics of products and explores the evolution of market structure in different customization levels. For the first time, we discuss the impact of customization proficiency on the competition between traditional software products and SaaS products and discuss the trade-off between customization requirements and economies of scale in the SaaS product line.

The structure of this paper is as follows: Sect. 2 reviews the relevant literature. Section 3 introduces our model. We show significant results in Sect. 4. Section 5 extends our model to analyze other technical aspects related to SaaS customization. First, we extend it to study the configuration customization techniques, which are similar to the personalization customization techniques. Second, we consider the modification customization techniques and make a comparison between the previous. Section 6 responds to the four research questions and discusses managerial implications based on the results. Finally, in Sect. 7, we address the limitations of the research and give further research directions.

[2] See more at: https://www.zdnet.com/article/enterprise-software-vendors-face-deflation-advantage-saas/.

2 Literature review

This work is relevant to the research fields of cloud computing and software versioning. Cloud computing is a resource pool integrated with many virtualized resources that are easy to use and access, such as hardware, development platforms and/or services (Vaquero et al. 2009). Due to the development of cloud computing technology and Web tools and thanks to the continuous reduction of bandwidth costs, the SaaS model has emerged. The SaaS concept can be traced back to Application Service Providers (ASPs) in the late 1990s (Ma and Seidmann 2015). Service providers dynamically set a variable load or scale of resources to realize optimal resource usage (Dutt et al. 2018). There is redundant literature of competition between the SaaS model and the traditional perpetual model, which we review in this part. We also focus on the researches of SaaS customization technologies in the following subsections.

2.1 On-premises software competes with SaaS software

Many researchers focus on competitive decisions between full-featured on-premises and limited-featured SaaS versions (Cheng et al. 2015; Choudhary 2007; Guo et al. 2018; Ma and Kauffman 2014; Shivendu and Zhang 2015). The fully functional version and the limited functional version are vertically differentiated products in nature. Mussa and Rosen (1978) study the pricing strategy of product lines and find that consumers with low valuation would not buy low-priced products. Because of the negative effect of product line strategy, monopoly software vendors charge high prices for low-quality products in order to extract more consumer surplus. Some researchers also find that low-quality products can compete with high-quality products and come to the same conclusion that consumers with low valuation would be priced out of buying (Moorthy 1984; Katz 1984). Varian (2000) constructs a selling and leasing model of information products and find that for information products that can be reused many times, manufacturers can benefit more from selling than from leasing. For which transaction cost is lower than the marginal production cost when sharing, leasing has more benefits than selling.

Bhargava and Choudhary (2008) build a stylized model and provide a research paradigm for information product versioning research. They reveal the conditions that different versions of information goods should be provided at the same time. Choudhary (2007) researches the impact of licensing models (SaaS or perpetual licensing) on software quality and indicates that high-quality SaaS brings higher profits and social welfare. Jia et al. (2018) study the profitability of two software licensing models under conditions of software updates and different price discrimination strategies. They investigate the conditions under which the selling model can be more profitable than the leasing model. Some researches pay attention to the network effect. According to Zhang and Seidmann (2010), when the network effect is large enough, software vendors should not transform into pure SaaS providers. Still, they should provide perpetual licensing and subscription models both. Based on network effects and SaaS product quality improvement, Guo and Ma (2018) study the competition between SaaS vendors and incumbent perpetual software vendors. They exhibit various pricing strategies that traditional software vendors may adopt to respond to SaaS competition.

However, none of the above researches captures the character of economies of scale, which is one of the critical features of the SaaS model. The economies of scale of the SaaS model benefit from cost-sharing and server load balancing (Ge and Huang 2014). The multitenant architecture of SaaS brings economies of scale, but also leads to consumer mismatch cost. Ma and Seidmann (2015) examine the dynamic competition between SaaS vendors and traditional software vendors. They reveal four different market structures that vary in mismatched costs. The paper suggests that SaaS providers can pass on the benefits of economies of scale to consumers to increase competitiveness by reducing consumer mismatch costs and price per transaction. But they do not directly address the trade-off between economies of scale and lack-of-fit cost and we discuss the trade-off in this work.

2.2 Software as a service customization

A large number of studies on enterprise applications have claimed that SaaS products lack personalization and provide standard common functionality (Ma and Kauffman 2014; Ma and Seidmann 2015; Li et al. 2018). However, due to the development of cloud computing technology, improved technologies and strategies make SaaS customization possible. A large scale of studies indicate that it is technically mature to carry out comprehensive SaaS customization.

Dutt et al. (2018) consider the modularity of the SaaS architecture and its performance when building a decision model for SaaS providers. They investigate the service design with respect to user preferences and relevant costs. Makki et al. (2018) analyse the different customization levels of SaaS applications, such as modifying user interfaces, changing databases, altering web services, and adjusting business process or workflow definitions. Ali et al. (2019) conduct a systematic literature review on SaaS customization and find that the current types of SaaS product customization can be divided into six categories: personalization, configuration, and modification as well as combination, integration, and extension. The latter three involves extensions with external service providers, and the combination of external programs, as well as the integration of external users. The other three can be divided into two categories. One type is pre-defined configuration and personalization (Sun et al. 2008; Mohamed et al. 2014); that is, users can only select, combine and configure from pre-set options. Another type involves source code alteration (Kong et al 2010; Moens and De Turck 2014; Ziani and AlShehri 2015).

According to Ali et al. (2019), most research of customization is about the configuration. Configurable customization techniques allow customers to configure and customize by themselves without changing the source code of the SaaS application running instance. Configurable customization is occurred at different levels, including Graphical User Interface (GUI), application logic and database logic. Changes in these levels will cause changes to the user interface, workflow, and business rules (Sun et al. 2008). Fan et al. (2015) point out ways to personalization customization

Table 1 The comparison of different customization techniques

	Personalization	Configuration	Modification
Data layer customization	√	√	√
Business flow layer customization	√	√	√
GUI layer customization	√	√	√
Alter source code base	×	×	√
Smart recommendation	√	×	×
Customization cost	Relatively high	Low	High
Economies of scale	Relatively high	High	Low

techniques. It takes steps that contain data collection, data preparation and transformation, data preservation, and recommendation, as well as personalization and service delivery. It integrates some existing customization methods and realizes multilayer customization based on ontology (Tsai et al. 2010). The differences between personalization customization and configuration customization are the processes such as intelligent recommendation in the user selection stage. Based on changing source code, modification customization can exceed the limits faced by configuration and personalization customization. Modifying source code is very costly because the vendor needs to pay higher to employees with higher skills, to allocate resources to manage different source codes (Sun et al. 2008). Here lists the similarities and differences of the three customization technologies in the following Table 1

3 Models

When building the model, we focus on a transitional vendor (such as SAP) that can provide both on-premises software and SaaS customized version from per transaction perspective (Ma & Seidmann, 2015). The previous study focus on competition between traditional software vendors and emerging SaaS providers. But this work concentrates on the cannibalization problem in the product lines. Besides, we examine the trade-off between customization proficiency and economies of scale. The previous study only mentions the impact of economies of scale on pricing but doesn't reveal the game between economies of scale and customization proficiency.

In subsection 3.1, we model the market where the monopoly software vendor only provides traditional software products, which is the benchmark model. In subsection 3.2, we build a hybrid product decision model for the vendor in the transition period. Model parameters refer to Table 2. For convenience, subscript *b* represents the model **b**efore transition and t represents the model in **t**ransition.

3.1 Model before transition

Small, medium, and big corporations have the IT needs to finish transactions in production and sales. Referring to the setting of Ma and Seidmann (2015), we

Table 2 Notations and definitions

Notes	Definitions
$d_i,\ d^*$	User expected transaction volume, undifferentiated user's location
θ	The fluctuation of IT demands volatility
u	The revenue of per-transaction
u_1, u_2	The utility of buying on-premises software, utility of buying SaaS
c_1	Service cost of operation and maintenance per transaction
c_2	Unit cost of IT infrastructure
c_3	SaaS customization cost
Q_i, Q_i^{1*}, Q_i^{2*}	User IT capacity; Installation capacity; Subscription capacity
D_i	The capacity of users' real IT needs
$p_1,\ p_2$	Traditional software price; Subscription fees per transaction
Π_b, Π_t	Profit **b**efore transition; Profits in **t**ransition
a	The maximum lack-of-fit cost of SaaS products and services
η	Vendors' customization ability or customization efficiency
t	The coefficient of economies of scale
w	The average cost per unit of vendor's cloud-deployed infrastructure

assume that every user is evenly distributed on [0, 1] with its expected trading volume evenly distributed on a line with a unit length of 1. The location of the consumer also represents the expected transaction volume.

Because of external factors such as market competition, the actual IT demands are volatile, and its fluctuation is $\theta\,(\theta < 0.5)$. Therefore, the actual transaction volume of consumers at the location d_i is $\left[d_i - \theta, d_i + \theta\right]$, and the actual transaction volume is a random variable that cannot be accurately estimated by users. Given the expected capacity of transactions, consumers need to install their own IT infrastructure to support them. The capacity is denoted by Q_i, and the unit architecture cost of the infrastructure is c_2. For each transaction, the revenues of the software product to the users are $u, (u > c_1 + c_2)$. However, in order to maintain the infrastructure and ensure the smooth operation of each transaction, the operation and maintenance service fees c_1 shall be paid per transaction.

When a user purchases traditional software, the vendor charges the user a one-time prepaid fee p_1, including royalties for the software product and customization fees for modifying the software code according to the user's needs. We assume that the software vendor provides only on-premises software in pre-transition period. We do not examine the marginal cost and fixed cost of traditional software (Cheng et al. 2015; Ma and Kauffman 2014).

When a user purchases a traditional software product, she will obtain the price through the marketing channel. Then she will decide whether to purchase the on-premises software and determine the capacity needed to support the expected trading volume. Software vendors need to make a decision on the price in advance, and then provide the software code to meet the personalized needs. Therefore, we use backward induction to solve the pricing problem. We first calculate user's installation capacity based on her expected trading volume, and then imply whether the user will buy the product, and finally get the optimal on-premises price.

It is difficult to estimate the users' IT capacity accurately due to unknown actual transaction volume such that their actual IT capacity is a random variable. For brevity, we neglect the capacity redundancy cost and deficiency cost as there is no significant impact on the main results. Therefore, the optimal capacity for users to install IT infrastructure is derived by Eq. (1). The first equation represents the expected net benefit from using the software product, and the second term represents the total cost of installing the IT architecture.

$$\max_{Q_i} E(u - c_1)\min\{Q_i, D_i\} - c_2 Q_i - p_1 \tag{1}$$

The expected utility function of the user is $Eu_1 = d_i(u - c_1) - c_2 Q_i^{1*} - p_1$, then the undifferentiated user is located at d^*. The profit function of traditional software suppliers is in Eq. (2).

$$\max_{p_1} \Pi_b = (1 - d^*)p_1 \tag{2}$$

3.2 Model in the transition period

When faced with hybrid products, new users brought by the new SaaS product line have two options: on-premises software and SaaS. If users choose traditional software, then they decide to install IT infrastructure capacity. If they choose SaaS, after purchasing products and services, the software vendor deploys the IT infrastructure capacity. Many SaaS providers rent cloud services (e.g., AWS) as their infrastructure.[3] It is essentially rent the IT infrastructure capacity and cloud-based services over the Internet that users subscribe to SaaS. SaaS users can calculate their SaaS capacity requirements Q_i^{2*} by Eq. (3). The first term in Eq. (3) means the total revenue from transactions, and the second term represents the total expenditure for subscribing capacities.

$$\max_{Q_i} \cdot u \min\{Q_i, D_i\} - p_2 Q_i \tag{3}$$

When the vendor provides a non-customized SaaS product, multitenant shares the same instance in a central location, leading to lack-of-fit costs for users. In order to reduce the mismatch cost, the software vendor introduces the customization technology to SaaS, and we measure their ability to customize by customization efficiency $\eta, (0 \leq \eta < 1)$. The software vendor has informed η to their potential SaaS consumers through advertising campaigns and information disclosure activities such as consulting services before buying. Therefore, the mismatch cost per transaction when the user buys a customized SaaS product is $a(1 - \eta)$. The better the SaaS is customized, the lower the mismatch cost is.

[3] See more at: https://aws.amazon.com/cn/.

The expected utility function of a user subscribing to a SaaS product is $Eu_2 = (u - p_2 - a(1-\eta))d_i$. By solving $Eu_1 = Eu_2$, we find that the indifferent user between purchasing licensed software and subscribing to SaaS products locates at $d^* = -\frac{\theta c_2 c_1 + 2\theta c_2^2 - \theta c_2 u + p_1 c_1 - p_1 u}{(-u+c_1)(a\eta - a + c_1 + c_2 - p_2)}$.

There are many technologies and theories of SaaS product customization, mainly including personalization, configuration, and composition as well as extension and modification and integration (Ali et al. 2019). Personalization is also referred as customization or mass customization. It uses flexible processes and organizational structures to achieve standardized mass production systems and produces personalized products and services at low costs (Gilmore and Pine 1997). Decreasing from high level to low level, the personalization customization occurs in the data layer, business process layer, and user interface layer. The personalization customization technology framework includes data preparation and pre-processing and data preservation and personalized service delivery (Fan et al. 2015). We assume the cost of personalization customization is the fixed cost with a linear form $F^P = c_3\eta$. In Sect. 5, we discuss different cost structures of different customization technologies to check the robustness of the model.

We can derive the profit of software vendors with hybrid product strategies by Eq. (4). The first term represents the total revenue of the customized SaaS. The second term represents the total infrastructure cost of SaaS, and the third term is customization cost. The last term shows the income from on-premises.

$$\max_{p_1,p_2} \Pi_t = \int_0^{d^*} (p_2 - c_1)x dx - twd^* Q^{2*} - \eta c_3 + (1-d^*)p_1 \tag{4}$$

$$\text{S.T.} \quad c_1 \le p_2 \le u - a(1-\eta); \quad p_1 \ge 0; \quad 0 \le d^* \le 1 \tag{5}$$

The vendor deploys the IT infrastructure based on the largest volume and makes the capacities huge enough to share in multitenant. The unit infrastructure cost of the vendor is denoted by w, and the infrastructure capacity is always subscribed from upstream cloud service providers such as AWS. The software vendor's economies of scale, denoted as t ($t < 1$), describes the software vendor's ability to coordinate in multiple tenants on shared operation and maintenance services. In other words,twd^*Q^{2*} represents the total infrastructure cost of software vendors under economies of scale. A smaller t means a bigger cost saving ability.

4 Results and analysis

4.1 The market before transition

After solving the simplified newsvendor problem presented by Eq. (1), we find the optimal capacity Q_i^{1*} for the focal user. From Eq. (6), we can see the impact of infrastructure cost and operation cost on capacity installation. The capacity decreases as operational costs c_2 and unit infrastructure costs c_1 increase.

$$Q_i^{1*} = d_i + \theta \frac{u - c_1 - 2c_2}{u - c_1} = d_i + \theta \left(1 - \frac{2c_2}{u - c_1} \right) \tag{6}$$

Then we get the optimal price of the on-premises software as $p_1^{'} = \frac{c_1^2 + \left((\theta+1)c_2 - 2u\right)c_1 + 2\theta c_2^2 - u(\theta+1)c_2 + u^2}{2u - 2c_1}$, by solving Eq. (2).

And the market share of the product is $1 - d^* = \frac{c_1^2 + \left((\theta+1)c_2 - 2u\right)c_1 + 2\theta c_2^2 - u(\theta+1)c_2 + u^2}{2\left(u - c_1 - c_2\right)\left(u - c_1\right)}$. We observe that fluctuations in IT demand had a significant impact on software vendors.

Proposition 1 (The effect of demand volatility) *If* $u < c_1 + 2c_2$, *then* $\frac{\delta(1-d^*)}{\delta\theta} > 0$; $\frac{\delta p_1^{'}}{\delta\theta} > 0$; $\frac{\delta\Pi_b^*}{\delta\theta} > 0$. *On the versus If* $u \geq c_1 + 2c_2$, *then* $\frac{\delta(1-d^*)}{\delta\theta} \leq 0$; $\frac{\delta p_1^{'}}{\delta\theta} \leq 0$; $\frac{\delta\Pi_b^*}{\delta\theta} \leq 0$

See appendix for proofs of all propositions and lemmas.

When users have relatively high per-transaction revenue ($u \geq c_1 + 2c_2$) from Eq. (2) $Q_i^{1*} = d_i + \theta\Delta$, then we get $\Delta > 0$. With an increase IT demand uncertainty, user installation capacity increases, so does the cost. As a result, undifferentiated consumer's location moves right on the hoteling line. Therefore, if software products bring higher revenue to users, less users may purchase the software products when faced demand uncertainty. Software vendors will reduce prices to gain more users, but the profit loss from price cuts dominates increased demand and lower software vendors' profits.

On the contrary, when the revenue is moderate ($c_1 + c_2 < u < c_1 + 2c_2$), and $\Delta < 0$. Software vendors benefit from increased IT demand volatility as the cost of installation for users decreases with increased IT demand uncertainty. Therefore, in the market with increased IT demand volatility, software vendors have an incentive to adjust the revenue per transaction to make it moderate. A reasonable revenue ($c_1 + c_2 < u < c_1 + 2c_2$) can not only enable users to use software products but also enable software vendors to make profits from the increased volatility.

4.2 The market in transition period

In the market before transition, the demand fluctuations obviously impact the decision-making of both users and software vendors. It is difficult for software vendors to make decisions within the estimation of demand fluctuations. However, on-demand ordering of SaaS will reduce the impact of demand fluctuations on supplier decision-making. But at the same time, providing hybrid products that may be competitive will cause product line cannibalization problems. During the transformation, we pay attention to the impact of customization efficiency on the cannibalization problem.

By solving the simplified newsvendor problem, we can derive the SaaS users' demand for IT capacity Q_i^{2*} (Eq. (7)), which decreases with the subscription fee and increases in the per-transaction revenue.

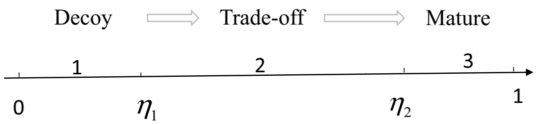

Fig. 2 Product decision road map

$$Q_i^{2*} = d_i + \theta\frac{u - 2p_2}{u} = d_i + \theta\left(1 - \frac{2p_2}{u}\right) \tag{7}$$

The user's optimal SaaS capacity depends on the revenue and price per transaction. If $u > 2p_2$, we have $\frac{\delta Q_i^{2*}}{\delta\theta} > 0$. That is to say, when revenue from per transaction is higher, the user orders more capacity to cope with demand fluctuations. On the contrary, if $u < 2p_2$, they order less to defend the demand uncertainty $\frac{\delta Q_i^{2*}}{\delta\theta} < 0$.

By solving Eq. (4), we find two important turning points of product strategy decision-making.

Proposition 2 *Two milestones of customization proficiency.*

$$\eta_1 = \frac{((-tw + c_2)\theta + a - c_2)u^2 + (((3tw - c_2)c_1 - 2c_2^2)\theta - c_1(a - c_2))u - 2\theta c_1^2 wt}{au(u - c_1)};$$
$$\eta_2 = \frac{4w(u - c_1)(u/2 + a - c_1 - c_2)t - uc_2(u - c_1 - 2c_2)c_2}{4wa(u - c_1)t}$$

when $\eta < \eta_1$, traditional products dominate the market, and if $\eta > \eta_2$, SaaS products dominate the market.

4.3 The effect of customization proficiency on market structure

4.3.1 Stage 1: Market under low customization efficiency

Moreover, there are three types of market structure possibly faced by vendors (see Fig. 2). When the software vendor has only a low level of customization $\eta < \eta_1$, traditional on-premises software dominants SaaS; when the customization level is moderate, that is $\eta_1 \leq \eta < \eta_2$, a hybrid product strategy segments market; when the customization level is high,$\eta_2 \leq \eta$, customized SaaS dominants the market.

Lemma 1 When customization efficiency is low, SaaS has a spillover effect on traditional software, and on-premises software dominates the market.

Under low customization capability ($\eta < \eta_1$), at stage 1, the prices are:

$$\left(p_1^* = -\frac{\theta c_2(c_1 + 2c_2 - u)}{-u + c_1}; p_2^* = \frac{u((c_1 - u)(tw\theta - a(1 - \eta) + c_2(1 - \theta) + c_1) - 2\theta c_2^2)}{(2tw\theta + u)(c_1 - u)}\right)$$

The profit of software vendors is $\Pi_t^{1*} = \frac{\theta c_2(c_1+2c_2-u)}{u-c_1} - c_3\eta$. The demand of SaaS products is $d_t^{1*} = 0$. There is a "decoy effect" that encourages more users to choose cheaper traditional on-premises software rather than inefficient customized SaaS products.

When the customization capability is at a low level, the low-transaction users who are supposed to subscribe to SaaS products turn to the on-premises for a reduced price $\left(p_1^* < p_1^{'}\right)$ and lower lack-of-fit costs of traditional products. At this time, the customized SaaS products play the role of "bait", driving consumers to choose on-premises software. However, although traditional software products occupy all the markets under this circumstance, its total profit is worse than pre-transition. Profit loss is due to that price decline dominates the increasing demand.

4.3.2 Stage 2: market under moderate customization efficiency

Lemma 2 *When customization proficiency is between the two milestones $\eta_1 \le \eta < \eta_2$, SaaS products compete with traditional software products, and the demand for SaaS products increases in customization ability.*

At stage 2 ($\eta_1 \le \eta < \eta_2$), SaaS products compete and substitute traditional products, the prices are

$$\left(p_1^*, p_2^*\right) = \left(\frac{A_1u^2 + B_1u - 2w\theta\left((\eta-1)a + c_2\right)tc_1^2}{2\left((\eta-1)a - tw + c_2\right)\left(u - c_1\right)u}\right), c_1$$

When a vendor offers both SaaS and traditional software consumption modes, users will choose between the two products based on their expected transaction volume. Since the lack-of-fit cost for users of SaaS products increases in expected volume, the consumer surplus of high-transaction users who buy SaaS products is relatively lower than those who buy traditional products. As a result, when expected transaction volume is lower than d_t^{2*} users choose SaaS model, and when it is higher than d_t^{2*}, users turn to traditional software products.

With the improvement of product customization proficiency, the consumer surplus increases as the lack-of-fit cost decreases, SaaS products compete with traditional software products and capture some users from the market of traditional software products. Thus, the demand for SaaS products increases as customization proficiency increases.

Proposition 3 *When the customization efficiency increases and exceeds a threshold, economies of scale becomes more complicated. The trade-off between economies of scale and customization proficiency makes a largest market share of SaaS.*

On the one hand, when the coefficient of economies of scale is smaller, the software vendors can coordinate the operation and maintenance among more tenants. It

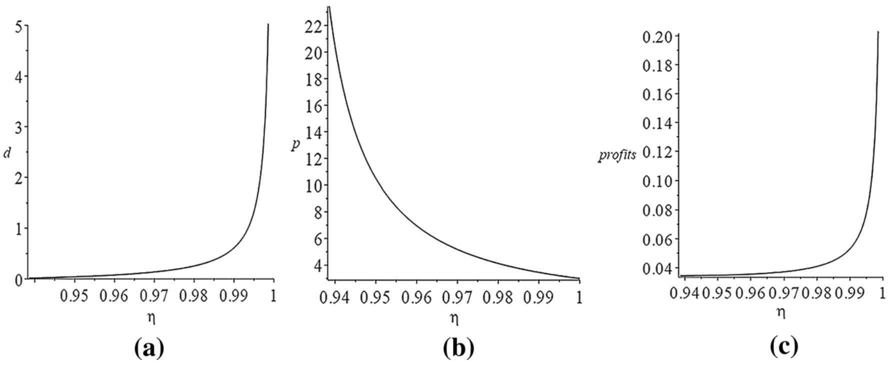

Fig. 3 **a** SaaS product demand versus customization proficiency, **b** SaaS price versus customization proficiency, **c** SaaS product profit versus customization proficiency

is helpful for the expansion of SaaS product demand. On the other hand, the software vendor provides products with high customization proficiency, which helps the expansion of SaaS product demand. However, with high customization proficiency, it is challenging to coordinate the maintenance services of multiple personalized tenants. Therefore, the market share of SaaS products is determined by the trade-off between economies of scale and customization proficiency.

We observe that there is a customization proficiency $\eta^*(\eta_1 < \eta^* < \eta_2)$. When the customization capability of software vendors is lower than η^*, i.e., $\eta_1 < \eta < \eta^*$, the software vendor has enough ability to support the increased customization demand. SaaS product demand increases with the coefficient of economies of scale and customization proficiency, i.e. $\frac{\delta d_t^{2*}}{\delta t} > 0$, $\frac{\delta d_t^{2*}}{\delta t \delta \eta} > 0$. When the customization efficiency is higher than η^*, we have $\frac{\delta d_t^{2*}}{\delta t} \leq 0$, $\frac{\delta d_t^{2*}}{\delta t \delta \eta} < 0$. At this time, it is difficult for the software vendor to coordinate with an increasing demand for customization, and the increase in economies of scale and customization efficiency will reduce the demand for SaaS products.

4.3.3 Stage 3: market under high customization efficiency

Lemma 3 *When $\eta_2 \leq \eta$, SaaS products can dominate the market.*

When a monopolistic vendor provides a high customization efficiency, prices are:

$$(p_1^*, p_2^*) = \left(0, \frac{A_2u^2 + B_2u + 4wtc_1\left(c_1 + (\eta - 1)a + c_2\right)^2}{-4tw\left(u - c_1\right)\left(\frac{-u}{2} + (\eta - 1)a + c_2 + c_1\right) - uc_2\left(u - c_1 - 2c_2\right)}\right).$$

We find that the perpetual software price is zero, and the software vendor cannot make any profit on that product. There is no competition between on-premises software and SaaS. The SaaS consumption model dominates the market.

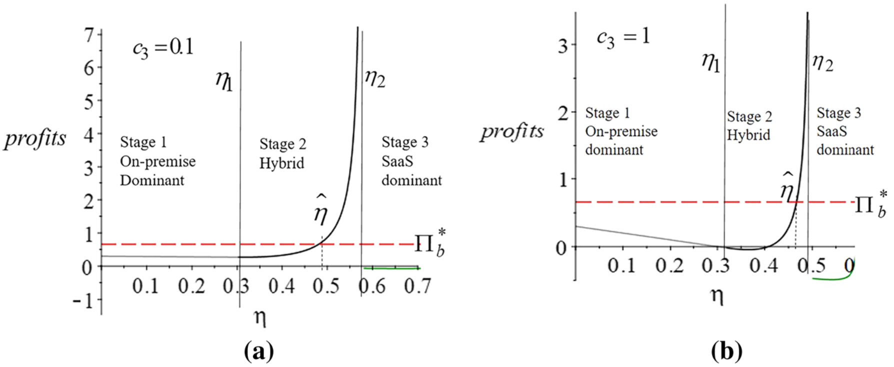

Fig. 4 **a** Profits of different product strategies with a low customization cost, **b** Profits of different product strategies with a high customization cost

In this case, due to the improvement of customization proficiency, the user's lack-of-fit cost is greatly reduced. Compared with traditional on-premises software, SaaS products will be the choice of the vast majority of users.

However, the price of SaaS products decreases in customization proficiency (Fig. 3(b)) and this is due to SaaS products try to capture the entire market. The profit increase brought by the increasing demand (Fig. 3(a)) dominants the profit loss by the price cut, so the profit of vendor still shows an upward trend in customization efficiency (Fig. 3(c)).

Proposition 4 *Optimal product strategy. We find* $\Pi_b^* > \Pi_t^{1*} > \Pi_t^{3*}$ *and* $\Pi_t^{2*} > \Pi_b^*$ *(under certain conditions). The competitive hybrid product strategy may generate the optimal profit for software vendors, but the SaaS dominated strategy will yield the worst profit.*

In stage 1, the product profit comes from traditional products. When traditional software reduces its price, although the demand increases, the profit still loses, and it is lower than that before the transformation. So it is not the best strategy for software vendors in the transformation period.

In stage 2, the profit increases in customization efficiency. When the customization proficiency meets certain conditions (larger than $\hat{\eta}$ in Fig. 4), the profit of hybrid products may exceed the profit before transition, reaching the optimal level. When SaaS products dominate the market, the profit from SaaS may be the worst.

In particular, whether the software vendor should provide customization services depends on the customization proficiency. When software vendors do not offer SaaS product customization services, i.e., customization efficiency $\eta = 0$ in Fig. 4, the vendor generates the optimal profit in stage 1. When the software vendor has moderate customization proficiency, providing customized services can bring the optimal profit, as long as the customization level is controlled at $[\hat{\eta}, \eta_2]$.

Besides, we compare the impact of different customization costs on the profits in Fig. 4(a) and (b). With an increase of customization cost ($c_3 = 1$ in Fig. 4(b)), the hybrid product strategy may bring negative profit, which shows the importance of controlling the input of customization. At the same time, the customization efficiency corresponding to the optimal profit moves left. That is to say, when the customization cost increases, the vendor may provide SaaS with a relatively lower level of customization.

5 Model extension and analysis

We now work on two extensions of the current model. We extend our model to two different customization technologies: configuration customization (in Sect. 5.1) and modification customization (in Sect. 5.2).

5.1 Model of configuration customization and analysis

The industry focus differences and organizational behaviours differences as well as operation strategy differences lead to various customization requirements for SaaS products. The widely used configuration customization technology supports different SaaS application variants with pre-defined parameters or setting tools to alter application functions within a pre-set scope. For example, users can add fields, change field names, and modify business rules.

It is worth noting that the configuration customization does not involve the modifying the code base such that the configuration can only support customization within the configurable limit. Moreover, various consumers may have more complex tailoring requirements for complicated software, which is often difficult to deliver through the web within multitenant architecture. That is why software with standard functions such as CRM and Human Resource (HRM) can achieve success in SaaS delivery model (Sun et al. 2008).

In this subsection, we work out the optimal pricing strategy and possible market structure for the software vendor who adopts configurable customization technology. Due to limited configurable customization, consumers still suffer mismatch cost $a(1-\eta^c)$, η^c is the configurable customization efficiency. For convenience, the superscript c stands for configuration. The utility of consumers who purchase the configurable SaaS is $Eu_2^c = (u - p_2^c - a(1-\eta))d_i$. Then, we calculate the demands of configurable SaaS d^{c*} by $Eu_2^c = Eu_1$.

On the side of software vendor, it needs fixed cost to provide configurable customization service. Salesforce.com builds a platform to achieve the configurable customization needs of consumers.[4] Salesforce.com provides two different mechanisms for users to customize and tailor their applications. One way is that consumers

[4] See more at: (available at Aug 15,2020) https://www.salesforce.com/cn/campaign/salesforce-my-way/.

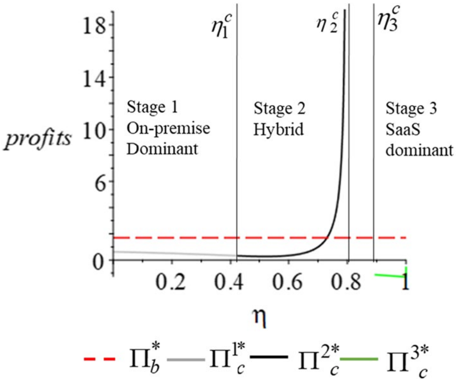

Fig. 5 Vendors' profits under different levels of configurable efficiency

use the tools provided to customize within the human–machine interaction interface. They can create data objects and modify the relationships between objects by themselves. The other method is to use the application programming interface provided by the aid of lightning components and Apex language programming. The fixed costs cover initial investment, and architecture design costs as well as technology research and development costs. We express the fixed cost as $F^c = \frac{1}{2}k(\eta^c)^2 + c_4\eta^c + m$ (Li et al. 2016), in which $k, m > 0$, $0 < c_4 < c_3$. This cost function is increasing and convex function in η^c.

During the transition period, the profit function of the software vendor who uses configuration customization technology to provide SaaS products is:

$$\max_{p_1^c, p_2^c} \Pi_t^c = \int_0^{d^{c*}} (p_2^c - c_1)xdx - twd^{c*}Q^{2*} - \eta^c c_4 - \frac{1}{2}k(\eta^c)^2 - m + (1 - d^{c*})p_1^c$$

S.T. $c_1 \le p_2^c \le u - a(1 - \eta^c)$; $p_1^c \ge 0$; $0 \le d^{c*} \le 1$.

We can get Q^{2*} from Eq. (3). By the same method mentioned in Sect. 4, we only show the results here. The equilibrium price and possible market structure depend on customization efficiency. We get four key points of configuration customization efficiency:

$$\eta_1^c = \frac{D_0u^2 + Du + 2twc_1^2\theta}{au(c_1 - u)}, \quad \eta_2^c = \frac{D_1u^2 + D_2u + 8tw\left(tw + a - c_1/4 - c_2\right)\theta c_1}{a\left(8c_1t\theta w - 8uw\theta t + 3c_1u - 3u^2\right)}$$

$$\eta_3^c = \frac{D_3u^2 + D_4u + 12tw\left(4/3tw + a - c_1/3 - c_2\right)\theta c_1}{4a\left(3c_1t\theta w - 3uw\theta t + c_1u - u^2\right)}, \quad \eta_4^c = 1 + \frac{2tw - c_2}{a}$$

.

Regime 1. $\eta^c < \eta_1^c$, traditional software serves market, and product prices are $p_1^{c*} = \frac{c_2\theta\left(c_1 + 2c_2 - u\right)}{u - c_1}$, $p_2^{c*} = \frac{u((c_1 - u)(tw\theta - a(1-\eta) + c_2(1-\theta) + c_1) - 2\theta c_2^2)}{(2tw\theta + u)(c_1 - u)}$;

Regime 2. $\eta_1^c < \eta < \eta_2^c$, SaaS competes with on-premises software. Products prices are $p_1^{c*} = \frac{A_3u^2+B_3u+2c_1^2+w\theta(c_2-a(1-\eta))}{2(c_1-u)u(a(\eta-1)-wt+c_2)}$, $p_2^{c*} = c_1$. The undifferentiated consumer is located at:

$$d^{c*} = \frac{\left(\left(-tw + c_2\right)\theta - c_2 + (-\eta + 1)a\right)u^2 + \left(\left(\left(3tw - c_2\right)c_1 - 2c_2^2\right)\theta + \left(c_2 + (\eta - 1)a\right)c_1\right)u - 2twc_1^2\theta}{2\left(-u + c_1\right)u\left(a\eta - tw - a + c_2\right)}$$

Regime 3. $\eta_3^c < \eta^c < \eta_4^c$, customized SaaS dominates the market, and the equilibrium price is:

$p_2^{c*} = \frac{A_2u^2+B_2u+4wtc_1\left(c_1+(\eta-1)a+c_2\right)^2}{-4tw\left(u-c_1\right)\left(\frac{-u}{2}+(\eta-1)a+c_2+c_1\right)-uc_2\left(u-c_1-2c_2\right)}$, the undifferentiated consumer who buys customized SaaS or not is located at:

$d^{c*} = -\frac{2\left(w\left(-u/2+(\eta-1)a+c_2+c_1\right)\left(-c_1+u\right)t+1/4uc_2\left(u-c_1-2c_2\right)\right)\theta}{u\left(-2tw+c_2+(\eta-1)a\right)\left(-c_1+u\right)}$.

It is worth mentioning that the pricing strategy of configurable custom technology is the same as personalization configuration technology. Under configurable technology, the level of customized SaaS that starts competing with On-premises is the same as that of personalization technology, i.e. $\eta_1^c = \eta_1$. Thus, under the configurable customization technology, the market structure may also show three dynamic evolutions: (1) traditional product on-premises software service market (2) customized SaaS compete with on-premises in the market (3) customized SaaS products dominate the market. However, vendor's profits vary in customization efficiency and we want to know what the optimal profit is. For intuitiveness, we compare the profits at transition period with the profit before the transformation by numerically simulating. Each parameter is set as $a = 5$,$u = 3$, $c_1 = 0.9$, $c_2 = 2$, $t = 0.5$, $\theta = 0.4$, $w = 2$, $c_4 = 0.5$, $k = 1$, $m = 0.1$.

Still, we find that it is not a choice for the vendor to occupy the market with highly customized SaaS products (Fig. 5). In stage 1, when the customization efficiency is relatively low, the profit is decreasing in efficiency. And it would be better for the vendor not to provide SaaS customization services. When the customization efficiency is moderate, the profit can break through that before the transformation. We briefly summarize this finding in Proposition 5.

Proposition 5 *Under configurable customization technology, the vendor can achieve optimal profit by serving hybrid products in which SaaS should be customized moderately.*

Configurable customization and personalization customization have the same impact on the market structure. With the improvement of customization efficiency by these two technologies, SaaS products finally have a solid market share. We still do not recommend that software vendors transform into pure cloud service providers with configurable techniques. Because the profit of single SaaS strategy may be much lower than that of hybrid product strategy. Software vendors' choice of customization technology during the transition period depends on the customization cost. When the configurable cost is lower than personalized cost, i.e. $F^c < F^p$, profit

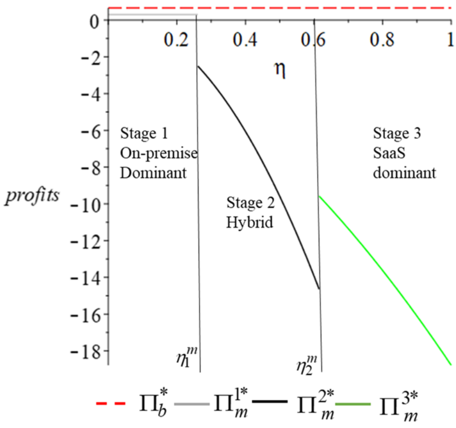

Fig. 6 The vendors' profit in different customization efficiency

under configurable technology is higher than that under personalization customization technology, i.e.$\Pi_t^{c*} > \Pi_t^{p*}$.

5.2 Model of modification customization and analysis

However, configurable customization and personalized customization may cause lack-of-fit costs for consumers because the mismatch between the SaaS variants and the required model for business rules (Jaroucheh et al. 2010). Tenants may also require SaaS vendors to modify the source code in the customization process (Moens and De Turck 2014). And modification customization methods are applied on a small scale (Ali et al. 2019).

The key feature of SaaS is that a large number of tenants share same SaaS architecture and database. The multitenant architecture allows software vendors to benefit from economies of scale. However, under the modification customization technology, the SaaS provider needs to manage every piece of customization code tenant by tenant (Sun et al. 2008). Thus leads to complexity in SaaS context and increasing customization investment. The vendor would suffer from impaired economies of scale. This part, we extend model to explain why major SaaS vendors are unwilling to develop and maintain software variants tenant by tenant? Under modification customization technology, what is the optimal pricing strategy for a monopolistic vendor? What are the possible market structures faced by the vendor?

Modifying the source code breaks through the limits of configuration and personalization customization. It can deeply meet the unique requirements of consumers. We assume that consumers no longer bear mismatch cost. We denote the customization efficiency of modification customization by η^m. For the sake of convenience, the superscript m stands for modification. The consumer utility of buying SaaS customized by modification techniques is $Eu_2^c = (u - p_2^m)d_i^m$. By solving $Eu_2^m = Eu_1$ we drive the demand for SaaS, that is $d^{m*} = -\frac{\theta c_2 c_1 + 2\theta c_2^2 - \theta c_2 u + p_1 c_1 - p_1 u}{(-u + c_1)(c_1 + c_2 - p_2)}$.

Although modification customization damages the economies of scale, we assume a well-designed infrastructure brings cost savings for the vendor. We denote the infrastructure cost as $C_I = twd^{m*}Q^{2*}$(we also consider that there are no cost savings in infrastructure investment, i.e.$C_I = wd^{m*}Q^{2*}$, but the main results still remain). The modification customization cost is the variable cost, denoted as $C_v = d^{m*}\eta^m c_3$. That is, customization costs increase in the number of tenants and customization efficiency. Under the modification customization technology, the profit of the software vendor is $\max_{p_1^m, p_2^m} \Pi_t^m = \int_0^{d^{m*}} (p_2^m - c_1)xdx + (1 - d^{m*})p_1^m - twd^{m*}Q^{2*} - d^{m*}\eta^m c_3$,

S.T. $c_1 \le p_2 \le u; p_1 \ge 0; 0 \le d^* \le 1$

After solving the equation, we find two key points of customization efficiency.

$$\eta_1^m = \frac{D_5u^2 + D_6u + c_1\left(8tw - 2c_1 - c_2\right)tw\theta}{c_3u\left(u - c_1\right)}; \eta_2^m = \frac{D_7u^2 + D_8u - c_1tw\theta\left(16tw - 4c_1 - 12c_2\right)}{2c_3u(c_1 - u)}$$

Still, we find that equilibrium prices and market structure depend on customization efficiency η^m.

Regime 1.$\eta^m < \eta_1^m$, the on-premises dominants market. The prices are $p_1^{m*} = \frac{c_2\theta\left(c_1+2c_2-u\right)}{u-c_1}$, $p_2^{m*} = \frac{u((c_2(1-\theta)+c_3\eta+tw\theta+c_1)(c_1-u)-2\theta c_2^2)}{(u+2tw\theta)(c_1-u)}$. The demand of SaaS is $d_1^{m*} = 0$.

Regime 2. $\eta_1^m < \eta^m < \eta_2^m$, on-premises software compete with the SaaS. The prices are $p_1^{m*} = \frac{c_2\left(A_3u^2+B_3u-2c_1^2tw\theta\right)}{-2u\left(w-c_2\right)\left(u-c_1\right)}$, $p_2^{m*} = c_1$. The demand of SaaS is:

$$d_2^{m*} = \frac{\left(\left(c_2 - w\right)\theta - c3\eta - c_2\right)u^2 + \left(\left(\left(3w - c_2\right)c_1 - 2c_2^2\right)\theta + c_1\left(c_3\eta + c_2\right)\right)u - 2\theta twc_1^2}{2u\left(tw - c_2\right)\left(u - c_1\right)}$$

Regime 3. $\eta^m > \eta_2^m$, customized SaaS expel the traditional software. We have $p_1^{m*} = 0$, $p_2^{m*} = \frac{A_4u^2+B_4u+4c_1\theta wt(c_1+c_2)^2}{4c_1w\theta t(c_1+c_2)+c_2\theta u(c_1+2c_2-u)-2u\theta wt(3c_1+2c_2-u)+2c_3u\eta(u-c_1)}$. The undifferentiated consumer who buys customized SaaS or not is located at: $d_3^{m*} = \frac{\left(\left(c_2-2tw\right)\theta-2c_3\eta\right)u^2+\left(\left(\left(-c_2+6wt\right)c_1+4twc_2-2c_2^2\right)\theta+2c_1c_3\eta\right)u-4twc_1\theta\left(c_1+c_2\right)}{4\left(u-c_1\right)\left(tw-c_2/2\right)u}$

Under different customization technologies, when SaaS products compete with traditional on-premises products, SaaS products adopt a forward-looking price strategy. Once SaaS products occupy the market, the price rises sharply. This forward-looking pricing strategy makes customized SaaS products popular in the SMEs market. But under the modification customization technology, the supplier's profit is not as good as the previous two customization technologies, and negative profits may appear (Fig. 6).

Proposition 6 *Under modification customization technology, the optimal strategy for the software vendor is not to provide customized SaaS products.*

Consumers still choose traditional products with lower prices when SaaS is customized at a low level. However, with the improvement of customization efficiency, SaaS begins to serve the market, and the market demand for SaaS products increases in customization efficiency, that is $\frac{\delta d_2^{m*}}{\delta \eta^m} = \frac{c_3}{2c_2-2tw} > 0$. But the price of SaaS only covers the maintenance costs, the revenue of on-premises needs to cover the customization costs. Moreover, the cost of SaaS customization outweighs the revenue from on-premises products, and the profit decreases in customization efficiency. Thus, the vendor may not provide modification customization SaaS products in the transition period.

6 Discussion and managerial implications

In reality, traditional software vendor faces many considerations when providing SaaS products, such as the time to enter the SaaS product market, the competition with existing SaaS providers, etc. Due to the complexity of the real environment, this paper discusses the impact of SaaS products on traditional software products and the impact of customization efficiency on pricing decisions. Different from the study of competitive software vendors, this paper finds some new enlightenment, which may guide the monopoly software vendor who offers a hybrid product strategy in transformation.

(1) The impact of SaaS product customization efficiency on vendor's decisions. We analyse common customization technologies and compare the optimal decisions of the software vendor under different customization technologies. We recommend that when the software vendor can provide a moderate customization efficiency, they can offer customized SaaS products, except for the case of modification customization technology. If the software vendor only has a low level of SaaS customization ability, they should offer SaaS products without customization. SaaS with an insufficient degree of customization cannot meet the basic needs of consumers. Consumers choose traditional software products with reduced prices.

On the contrary, when the vendor has a high level of customization, if highly customized products provided, the on-premises may be expel out of the market. But the profit of the vendor is not ideal in this case because software companies have to pay high customization costs to provide deeply customized products. Especially in the modification customization technology, SaaS customization needs source code alteration services to achieve in-depth customization. But this comprehensive customization technology requires people with higher skills and higher wages, and requires the allocation of resources and infrastructure to coordinate and manage different Software code versions. The optimal hybrid product strategy is different from the prediction of Varian (2000). Varian (2000) points out that leasing software will dominate the market when there is no quality difference between leasing software and licensing. Although high customization will reduce the quality difference between leasing software and licensing software, the non-negligible customization cost and decreased economies of scale do not allow software companies to adopt only leasing mode.

We suggest that software vendors make the customization proficiency moderate to capture the SMEs, and ensure that users with large trading volume rely on traditional software products.

(2) The effect of customization on vendor's transition decisions
Under a moderate level of customization efficiency, the SaaS products may take a fixed penetration price which only covers the marginal cost. This pricing returns the cost savings brought by economies of scale to consumers and wins a stable market share for SaaS products (Ma and Seidmann 2015). Large scale users are still the primary profit source for the software vendor. With the increase in customization efficiency, SaaS products fit more individual requirements. However, due to security and robustness needs, some large scale users still choose traditional software products. Therefore, the on-premises software price is increasing in customization efficiency and the vendor fully squeezes the large-scale consumer surplus to achieve an increasing profit. When the customization efficiency is extremely high, the software vendor only benefit from SaaS products. But the profit of a single SaaS strategy is inferior to that of the hybrid product strategy. Therefore, we do not recommend that the vendor fully transits to a pure cloud provide.
(3) Managerial implications
From a theoretical perspective, we contribute to the cloud computing research field by studying the impact of SaaS customization on versioning and fill the gap between theory and practice. This work studies the impact of customization efficiency on the dynamic competition for the first time. Secondly, we reveal the trade-off between SaaS customization and economies of scale. We provide a robust framework to analyse the impact of customization efficiency under different customization technologies on software vendors' decisions. Finally, we enrich the software versioning research and analyse the market structure evolution in the case of SaaS customization. We claim that the hybrid product strategy may be the best strategy for the vendor in a transitional period in which SaaS should be customized moderately.
From the perspective of practice, we find out the optimal pricing strategy and product strategy for the software vendor. Traditional on-premises software may reduce prices during the transition period. In this way, the low-level customized SaaS drives consumers to choose traditional software products. We recommend that the software vendor should improve their customization capabilities and their ability of economies of scale. The software vendor should control the technological improvement and should maintain the customization standards at a moderate level. We also suggest that the software vendor builds a unified platform and design a common architecture for different variants. By comparing different customization technologies, we find that economies of scale is an important profitable feature of the SaaS model. We use stylized models combined with numerical simulation to explain why modification is not favoured by major software vendors. Because the cost of modification customization increases in both demands and the degree of customization. Besides, we advise the vendor provides customization services for SaaS products with the characteristics of low complexity. For complex applications such as ERP systems, they are more suitable for local deployment. Under the current customization technology, it is costly to meet the individual needs of enterprises; secondly, it is difficult for the vendor to benefit from economies of scale by adopting high-level customization technology.

7 Conclusion

This work studies the product strategies of a monopoly software vendor in the transition period. Moreover, it reveals the impact of SaaS customization on the market structure, which breaks through the limitations of the non-customization in software versioning. When the software vendor adopts a hybrid product strategy in the transition period, the software vendor may use a forward-looking pricing strategy to help SaaS products gain a stable position in the market. The trade-off between customization and economies of scale affects the market share of SaaS. We believe that though SaaS products can be deeply customized, they will not replace traditional on-premises software products. The software vendor should not abandon perpetual software and become pure cloud service providers. Besides, the vendor should consider personalization or configuration customization technologies rather than modification customization techniques.

In particular, we find two milestones of customization efficiency that change market structure in the transformation period. When the customization proficiency is between the two milestones, SaaS customization products show the competition effect on the on-premises software. Besides, the trade-off between economies of scale and customization efficiency makes SaaS products the largest market share. Software vendors need to control the number of tenants and ensure economies of scale. Before reaching the first milestone of customization proficiency, SaaS products play a role of bait and drive users to on-premises market. We also find that SaaS products should take a forward-looking pricing strategy. This pricing strategy wins a solid market share for SaaS products and makes it possible to extract consumer surplus from large-scale users fully.

Finally, there are many limitations of this paper. We only consider the decision-making problems faced by monopolistic software vendors. We did not examine the issues of competing software vendors during the transition period. Companies such as IBM and Oracle have also adopted a hybrid product strategy. They provide both SaaS products and on-premises products. How should competing software vendors make pricing and customization decisions? Also, we assume that consumers only choose one version. When do consumers use hybrid products, that is, purchase SaaS and on-premises products at the same time? Besides, we only check the three popular types of customization technologies. How about the other three types that involves extensions with external service providers or applications? Future research may overcome the limitations of the current model by solving the listed questions. Besides, we did not investigate the switch of users who purchase on-premises software before transition and buy SaaS during the transition period. This problem is interesting and we refer the research of Guo and Ma (2018) for that problem.

Acknowledgements The authors thank the editor and reviewers for constructive advice during the review process and for nudging them toward an extension analysis of various customization technologies.

Funding This research is supported by the National Social Sciences Fund of China [grant number 15ZDB150] and National Natural Science foundation of China [Grant 71631003].

Table 3 All the combination of λ

	1	2	3	4
1	(1,0,0,0)	(0,1,0,0)	(0,0,1,0)	(0,0,0,1)
2	(1,1,0,0)	(1,0,1,0)	(1,0,0,1)	(0,1,1,0)
3	(0,1,0,1)	(1,1,1,0)	(1,1,0,1)	(1,1,1,1)
4	(0,0,1,1)	(0,1,1,1)	(1,0,1,1)	(0,0,0,0)

Table 4 The result of Lagrange problem

	1	2	3	4
1	Conditional solution 1 (demand of both)	No solution	No solution	Conditional solution 2 (only demand of SaaS)
2	Irrational solution $\lambda_1 = 0$	Irrational solution $\lambda_1 < 0$	Inferior solution	No solution
3	No solution	No solution	No solution	No solution
4	Irrational solution $\lambda_4 < 0$	No solution	No solution	Conditional solution 3 (only demand of on-premises)

Appendix

Proof of Proposition 1

The optimal price of on-premises software before transition is $p_1^{'} = \frac{c_1^2+((\theta+1)c_2-2u)c_1+2\theta c_2^2-u(\theta+1)c_2+u^2}{2u-2c_1}$ and the market share of traditional software is $1-d^* = \frac{c_1^2+((\theta+1)c_2-2u)c_1+2\theta c_2^2-u(\theta+1)c_2+u^2}{2(u-c_1-c_2)(u-c_1)}$, then we derive the profit of vendor by $\Pi_1^* = (1-d^*)p_1^{'} = \frac{(c_1^2+((\theta+1)c_2-2u)c_1+2\theta c_2^2-u(\theta+1)c_2+u^2)^2}{4(u-c_1)^2(u-c_1-c_2)}$.

Then we take the derivative of those with respect to θ, and we get. $\frac{\delta\Pi_1}{\delta\theta} = -\frac{\left(2\theta c_2^2-(\theta+1)(u-c_1)c_2+(u-c_1)^2\right)c_2(-c_1-2c_2+u)}{2(u-c_1)^2(u-c_1-c_2)}$ $\frac{\delta(1-d^*)}{\delta\theta} = \frac{c_2(c_1+2c_2-u)}{2(u-c_1-c_2)(u-c_1)}$; $\frac{\delta p_1^{'}}{\delta\theta} = \frac{c_2(c_1+2c_2-u)}{2u-2c1}$. Recall that $u > c_1 + c_2$, then if $c_1 + c_2 < u < c_1 + 2c_2$, we have $\frac{\delta\Pi_1^*}{\delta\theta} > 0, \frac{\delta(1-d^*)}{\delta\theta} > 0, \frac{\delta p_1^{'}}{\delta\theta} > 0$ and vice versa.

Proof of position 2

The software vendor's decision problem under the hybrid product strategy is

$$\max_{p_1,p_2} \Pi_2 = \int_0^{d^*} (p_2 - c_1)xdx - twd^*Q^{2*} + (1-d^*)p_1$$

$$\text{S.T. } c_1 \le p_2 \le u - a(1-\eta);\ p_1 \ge 0;\ 0 \le d^* \le 1$$

After simplifying the constraints we can write the Lagrange equation as

$$\Pi_2 = \int_0^{d^*} (p_2 - c_1)xdx - twd^*Q^{2*} - \eta c_3 + (1-d^*)p_1 + \lambda_1\left(p_2 - c_1\right) + \lambda_2\left(\frac{\theta c_1 c_2 + 2\theta c_2^2 - \theta c_2 u}{u - c_1} - p_1\right)$$
$$+\lambda_3\left(\frac{ac_1\eta - a\eta u + \theta c_1 c_2 + 2\theta c_2^2 - \theta c_2 u - ac_1 + au + c_1^2 + c_2 c_1 - c_1 u - c_2 u}{-u + c_1} - p_2 + p_1\right) + \lambda_4 p_1$$

There are 16 combinations of lambda and we summary all in the following table. For each lambda, it has two states. The first state is λ_i=0, and in the second state is $\lambda_i > 0(i = 1, 2, 3, 4.)$. We denote state 1 as 0 and state 2 as 1. In this case, λ_1=0, λ_2=0, λ_3=0, λ_4=0 can be denoted as (0, 0, 0, 0); and $\lambda_1 > 0$, λ_2=0, λ_3=0, λ_4=0 is written as (1, 0, 0, 0) Table 3 and 4.

We begin with combination (1,1), then we discuss combination (1,4) and finally we explain combination (4,4).

Conditional Solution 1

When $\lambda_1 > 0, \lambda_2 = 0, \lambda_3 = 0, \lambda_4 = 0$, there is demand of both on-premises software and SaaS. Thus we have.

$p_1^* = \frac{A_1 u^2 + B_1 u - 2w\theta\left((\eta-1)a + c_2\right)tc_1^2}{2\left((\eta-1)a - tw + c_2\right)\left(u - c_1\right)u}$ and $p_2^* = c_1$

$$d_t^{2*} = \frac{\left((wt - c_2)\theta + (\eta-1)a + c_2\right)u^2 + \left(\left((-3wt + c_2)\theta - c_2 + (-\eta+1)a\right)c_1 + 2\theta c_2^2\right)u + 2c_1^2 t\theta w}{2\left((\eta-1)a - wt + c_2\right)\left(u - c_1\right)u}$$

$$A_1 = -(\eta-1)^2 a^2 - \left((\theta+2)c_2 + \theta tw\right)(\eta-1)a + \left((-\theta-1)c_2 + \theta tw\right)c_2$$

$$B_1 = c_1(\eta-1)^2 a^2 + 3\left(\frac{2}{3}\theta c_2^2 + \frac{1}{3}c_1(\theta+2)c_2 + c_1 t\theta w\right)(\eta-1)a$$
$$+\left(2\theta {c_2}^2 + \left(\left(-4tw + c_1\right)\theta + c_1\right)c_2 + c_1 t\theta w\right)c_2$$

If $\eta > \eta_1 = \frac{\left((-tw+c_2)\theta + a - c_2\right)u^2 + \left(\left((3tw - c_2)c_1 - 2c_2^2\right)\theta - c_1\left(a - c_2\right)\right)u - 2\theta c_1^2 wt}{au\left(u - c_1\right)}$, we have $\lambda_1 > 0$.

Conditional solution 2

(See combination (1,4)) when $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0, \lambda_4 > 0$ there is only demand of SaaS.

If $\eta > \eta_2 = \frac{4w(u-c_1)(u/2+a-c_1-c_2)t-uc_2(u-c_1-2c_2)c_2}{4wa(u-c_1)t}$, we have $\lambda_4 > 0$. And the prices are.

$p_2^* = \frac{A_2u^2+B_2u+4wtc_1(c_1+(\eta-1)a+c_2)^2}{-4tw(u-c_1)\left(\frac{-u}{2}+(\eta-1)a+c_2+c_1\right)-uc_2(u-c_1-2c_2)}$ and $p_1^* = 0$.

Then $d_t^{3*} = -2\frac{(w(-u/2+(\eta-1)a+c_2+c_1)(u-c_1)t+1/4uc_2(u-c_1-2c_2))\theta}{u(-2tw+c_2+(\eta-1)a)(u-c_1)}$

$$A_2 = 2\left((\eta-1)a + c_1 - c_2\right)wt + \left(-c_1 + (\eta-1)a + c_2\right)c_2$$

$$B_2 = -4w\left((\eta-1)^2a^2 + \frac{2}{5}(\eta-1)\left(c_1 + \frac{4}{5}c_2\right)a + \frac{3}{2}c_1^2 + \frac{3}{2}c_1c_2 - c_2^2\right)t - \left(-c_1 + (\eta-1)a + c_2\right)c_2\left(c_1 + 2c_2\right)$$

Conditional Solution 3

When $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0, \lambda_4 = 0$ we derive a conditional solution with only demand of on-premises software (see combination (4, 4)). And we have $p_2^* = c_1$ and $p_1^* = -\frac{c_2\theta(c_1+2c_2-u)}{-u+c_1}$, $d_2^{1*} = 0$. Then $Eu_2 = (u - c_1 - a(1-\eta))d_i < 0$, thus $\eta < \eta_3 = 1 - \frac{u-c_1}{a}$, then we have $\eta_3 > \eta_1$. Thus when $\eta < \eta_3$,$Eu_2 = (u - c_1 - a(1-\eta))d_i < 0$ and there is only demand of on-premises software. So if $\eta < \eta_1$ there is only on-premises software dominated the market and if $\eta > \eta_2$ we have only demand of SaaS software, then if $\eta > \eta_1$ there is both demand of on-premises software and SaaS.

Proof of Lemma 1

According to proposition 2 if $\eta < \eta_1$ there is only on-premises software dominated the market.

Proof of Lemma 2

According to proposition 2 if $\eta_1 < \eta < \eta_2$ we have both demand of SaaS software.

$$d_t^{2*} = \frac{\left(\left(wt - c_2\right)\theta + (\eta-1)a + c_2\right)u^2 + \left(\left(\left(-3wt + c_2\right)\theta - c_2 + (-\eta+1)a\right)c_1 + 2\theta c_2^2\right)u + 2c_1^2 t\theta w}{2\left((\eta-1)a - wt + c_2\right)\left(u - c_1\right)u}$$

$$\frac{\delta d_t^{2*}}{\delta\eta} = -\frac{a\Big(\left(\left(tw - c_2\right)\theta + tw\right)\left(c_1 + 2c_2\right)^2 + \left(\left(-3twc_1 + c_2\left(c_1 + 2c_2\right)\right)\theta - twc_1\right)\left(c_1 + 2c_2\right) + 2c_1^2 t\theta w\Big)}{4c_2\left((\eta-1)a - tw + c_2\right)^2\left(c_1 + 2c_2\right)c_2} > 0$$

Thus the demand of SaaS increases in customization proficiency.

Proof of Proposition 3

Recall the proof of proposition 2, we can see the prices of on-premises software and SaaS.

$$\left(p_1^*,\ p_2^*\right) = \left(\frac{A_1u^2 + B_1u - 2w\theta\left((\eta-1)a + c_2\right)tc_1^2}{2\left((\eta-1)a - tw + c_2\right)\left(u - c_1\right)u};c_1\right)$$

$$A_1 = -(\eta-1)^2a^2 - \left((\theta+2)c_2 + \theta tw\right)(\eta-1)a + \left((-\theta-1)c_2 + \theta tw\right)c_2$$

$$\begin{aligned} B_1 = {} & c_1(\eta-1)^2a^2 + 3\left(\frac{2}{3}\theta c_2^2 + \frac{1}{3}c_1(\theta+2)c_2 + c_1t\theta w\right)(\eta-1)a \\ & + \left(2\theta {c_2}^2 + \left(\left(-4tw + c_1\right)\theta + c_1\right)c_2 + c_1t\theta w\right)c_2 \end{aligned}$$

Thus, the SaaS product line has an effect on on-premises product line by impacting the prices through customization proficiency η.

$$\eta^*\Big|_{\frac{\delta d_t^{2*}}{\delta t}=0} = \frac{\left(a\theta + a - c_2\right)u^2 + \left(\left(\left(-3a + 2c_2\right)c_1 - 2c_2^2\right)\theta - c_1\left(a - c_2\right)\right)u + 2c_1^2\theta\left(a - c_2\right)}{\left(u - c_1\right)\left((\theta+1)u - 2c_1\theta\right)a}$$

If $c_1 + 2c_2 > u > c_1 + c_2$, we have $\eta_1 < \eta^* < \eta_2$; when $\eta_1 < \eta < \eta^*$, thus $\frac{\delta d_t^{2*}}{\delta t} > 0$; and when $\eta^* \le \eta < \eta_2$, we have $\frac{\delta d_t^{2*}}{\delta t} \le 0$, $\frac{\delta d_t^{2*}}{\delta t\delta\eta} < 0$.

Proof of Lemma 3

$\eta_2 \le \eta < 1$, the optimal pricing strategy is.

$$\left(p_1^*, p_2^*\right) = \left(0,\ \frac{A_2u^2 + B_2u + 4wtc_1\left(c_1 + (\eta-1)a + c_2\right)^2}{-4tw\left(u - c_1\right)\left(\frac{-u}{2} + (\eta-1)a + c_2 + c_1\right) - uc_2\left(u - c_1 - 2c_2\right)}\right)$$

And the demand is.

$$d^* = -2\frac{\left(w\left(-u/2 + (\eta-1)a + c_2 + c_1\right)\left(u - c_1\right)t + 1/4uc_2\left(u - c_1 - 2c_2\right)\right)\theta}{u\left(-2tw + c_2 + (\eta-1)a\right)\left(u - c_1\right)}.$$

$$A_2 = 2\left((\eta-1)a + c_1 - c_2\right)wt + \left(-c_1 + (\eta-1)a + c_2\right)c_2$$

Table 5 The result of lagrange problem

	1	2	3	4
1	Conditional solution 1 (demand of both)	No solution	No solution	Conditional solution 2 (only demand of SaaS)
2	Irrational solution $\lambda_1 = 0$	(1,0,1,0)	(1,0,0,1)	(0,1,1,0)
3	(0,1,0,1)	(1,1,1,0)	(1,1,0,1)	(1,1,1,1)
4	(0,0,1,1)	(0,1,1,1)	(1,0,1,1)	Conditional solution 3 (only demand of on-premises)

$$B_2 = -4w\Big((\eta-1)^2a^2 + \frac{2}{5}(\eta-1)\Big(c_1 + \frac{4}{5}c_2\Big)a + \frac{3}{2}c_1^2 + \frac{3}{2}c_1c_2 - c_2^2\Big)t \\ - \big(-c_1 + (\eta-1)a + c_2\big)c_2\big(c_1 + 2c_2\big)$$

Proof of proposition 5

The profit function of the vendor is:

$$\max_{p_1^c, p_2^c} \Pi_t^c = \int_0^{d^{c*}} (p_2^c - c_1)xdx - twd^{c*}Q^{2*} - \eta^c c_4 - \frac{1}{2}k(\eta^c)^2 - m + (1-d^{c*})p_1^c$$

S.T.$c_1 \le p_2^c \le u - a(1-\eta^c)$; $p_1^c \ge 0$; $0 \le d^{c*} \le 1$.
After simplifying the constraints we can write the Lagrange equation as

$$\Pi_t^c = \int_0^{d^{c*}} (p_2^c - c_1)xdx - twd^{c*}Q^{2*} - \eta^c c_3 + (1-d^{c*})p_1 + \lambda_1\big(p_2^c - c_1\big) + \lambda_2\left(\frac{\theta c_1c_2 + 2\theta c_2^2 - \theta c_2 u}{u - c_1} - p_1^c\right) \\ +\lambda_3\left(\frac{ac_1\eta - a\eta u + \theta c_1c_2 + 2\theta c_2^2 - \theta c_2 u - ac_1 + au + c_1^2 + c_2c_1 - c_1u - c_2u}{-u + c_1} - p_2^c + p_1^c\right) + \lambda_4 p_1^c$$

Then we get $\eta_1^c = \frac{D_0u^2 + Du + 2twc_1^2\theta}{au(c_1-u)}$, $\eta_2^c = \frac{D_1u^2 + D_2u + 8tw\big(tw + a - c_1/4 - c_2\big)\theta c_1}{a\big(8c_1t\theta w - 8uw\theta t + 3c_1u - 3u^2\big)}$,
$\eta_3^c = \frac{D_3u^2 + D_4u + 12tw\big(4/3tw + a - c_1/3 - c_2\big)\theta c_1}{4a\big(3c_1t\theta w - 3uw\theta t + c_1u - u^2\big)}$, $\eta_4^c = 1 + \frac{2tw - c_2}{a}$.

$$D_0 = \big(\big(tw - c_2\big)\theta - a + c_2\big)$$

$$D = \big(\big(\big(-3tw + c_2\big)c_1 + 2c_2^2\big)\theta + c_1\big(a - c_2\big)\big)$$

$$D_1 = \big(\big(-tw + c_2\big)\theta - 4tw - 3a + 3c_2\big), \\ D_2 = \big(\big(-8t^2w^2 - 8\big(a - 3/8c_1 - c_2\big)wt - c_2\big(c_1 + 2c_2\big)\big)\theta + 3\big(4/3tw + a - c_2\big)c_1\big)$$

$$D_3 = \left(\left(-2tw + c_2\right)\theta - 8tw - 4a + 4c_2\right)$$

$$D_4 = \left(\left(-16t^2w^2 - 12\left(a - c_1/2 - c_2\right)wt - c_2\left(c_1 + 2c_2\right)\right)\theta + 4c_1\left(2tw + a - c_2\right)\right)$$

Conditional solution 1

$\eta_1^c < \eta < \eta_2^c$, there are demands from both SaaS and on-premises.

$$p_2^{c*} = c_1$$

$$p_1^{c*} = \frac{A_3u^2 + B_3u - 2tw\left(c_2 + (\eta - 1)a\right)c_1^2\theta}{2(c_1 - u)u(a(\eta - 1) - wt + c_2)};$$
$$A_3 = -(\eta - 1)^2a^2 - (\eta - 1)\left(\left(tw + c_2\right)\theta + 2c_2\right)a + c_2\left(\left(tw - c_2\right)\theta - c_2\right)$$

$$\begin{aligned} B_3 = {} & c1(\eta - 1)^2a^2 + 3(\eta - 1)\left(\left(c_1tw + 1/3c_1c_2 + 2/3c_2^2\right)\theta + 2/3c_1c_2\right)a \\ & + c_2\left(\left(2c_2^2 + \left(-4tw + c_1\right)c_2 + c_1tw\right)\theta + c_1c_2\right) \end{aligned}$$

Conditional solution 2

$$\eta_3^c < \eta^c < \eta_4^c$$

There is only demand from SaaS.
$p_1^{c*} = 0$

$$p_2^{c*} = \frac{A_4u^2 + B_4u + 4wc_1t(c_1 - c_2 + a(\eta - 1))^2}{-4\left(u - c_1\right)\left(-u/2 + (\eta - 1)a + c_2 + c_1\right)wt - uc_2\left(u - c_1 - 2c_2\right)}$$

$$A_4 = 2w\left((\eta - 1)a + c_1 - c_2\right)t + c_2\left(-c_1 + (\eta - 1)a + c_2\right),$$

$$\begin{aligned} B_4 = {} & -4w\left((\eta - 1)^2a^2 + 5/2(\eta - 1)\left(c_1 + 4/5c_2\right)a + 3/2c_1^2 + 3/2c_1c_2 - c_2^2\right)t \\ & -c_2\left(c_1 + 2c_2\right)\left(-c_1 + (\eta - 1)a + c_2\right) \end{aligned}.$$

Conditional Solution 3

When $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0, \lambda_4 = 0$ we derive a conditional solution with only demand of on-premises software. And we have.

$$p_1^{c*} = -\frac{c_2\theta\left(c_1 + 2c_2 - u\right)}{-u + c_1};$$

$$p_2^{c*} = \frac{u((c_1 - u)(tw\theta - a(1-\eta) + c_2(1-\theta) + c_1) - 2\theta c_2^2)}{(2tw\theta + u)(c_1 - u)}$$

We refer to the numerical simulation for the comparison of profits.

Proof of proposition 6

The profit of the software vendor is Table 5.

$\max_{p_1^m, p_2^m} \Pi_t^m = \int_0^{d^{m*}} (p_2^m - c_1)xdx + (1 - d^{m*})p_1^m - twd^{m*}Q^{2*} - d^{m*}\eta^m c_3,$ S.T. $c_1 \le p_2 \le u$; $p_1 \ge 0$; $0 \le d^* \le 1$. And the Lagrange equation is:

$$\Pi_2 = \int_0^{d^*} (p_2 - c_1)xdx - wd^*Q^{2*} - d^*\eta c_3 + (1-d^*)p_1 + \lambda_1(p_2 - c_1) + \lambda_2\left(\frac{\theta c_1 c_2 + 2\theta c_2^2 - \theta c_2 u}{u - c_1} - p_1\right)$$
$$+\lambda_3\left((-u + c1)p2 + p1(u - c1) + (u - c1)(c1 + c2) - \theta c2c1 - 2\theta c2^2 + \theta c2u\right) + \lambda_4 p_1$$

We get two key points of efficiency:

$$\eta_1^m = \frac{D_5u^2 + D_6u + c_1\left(8w - 2c_1 - c_2\right)w\theta}{c_3u\left(u - c_1\right)}$$

$$\eta_2^m = \frac{2c_1^2w\theta + c_1c_2\theta u - 3c_1\theta uw + 2c_2^2\theta u - c_2\theta u^2 + \theta u^2w - uc_2c_1 + u^2c_2}{c_3u\left(-u + c_1\right)}$$

$D_5 = \left(\left(-w + c_2\right)\theta - 4w + 3c_2\right)$,
$D_6 = \left(\left(-8w^2 + \left(3c_1 + 8c_2\right)w - c_2\left(c_1 + 2c_2\right)\right)\theta + \left(4w - 3c_2\right)c_1\right).$

Conditional Solution 1

When $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0, \lambda_4 = 0$ we derive a conditional solution with only demand of on-premises software. And the prices are $p_1^{m*} = \frac{c_2\theta\left(c_1 + 2c_2 - u\right)}{u - c_1}$, $p_2^{m*} = \frac{u((c_2(1-\theta) + c_3\eta + tw\theta + c_1)(c_1 - u) - 2\theta c_2^2)}{(u + 2tw\theta)(c_1 - u)}$. The demand of SaaS is $d_1^{m*} = 0$.

Conditional solution 2

$\eta_1^m < \eta^m < \eta_2^m$, on-premises software compete with the SaaS. The prices are $p_1^{m*} = \frac{c_2\left(A_3u^2 + B_3u - 2c_1^2tw\theta\right)}{-2u\left(w - c_2\right)\left(u - c_1\right)}, p_2^{m*} = c_1$

$A_3 = \left(-(\theta + 1)c_2 + \theta w - c3_3\eta^m\right)$,

$B_3 = \left(2c_2^2\theta + \left(\left(c_1 - 4w\right)\theta + c_1\right)c_2 + c_1\left(c_3\eta^m + \theta w\right)\right)$

Conditional solution 3

$\eta^m > \eta_2^m$, customized SaaS expel the traditional software. We have $p_2^{m*} = \frac{A_4u^2+B_4u+4c_1\theta wt(c_1+c_2)^2}{4c_1w\theta t(c_1+c_2)+c_2\theta u(c_1+2c_2-u)-2u\theta wt(3c_1+2c_2-u)+2c_3u\eta(u-c_1)}$.

$p_1^{m*} = 0.$

$$A_4 = (c_2 - 2w)(c_2 - c_1)\theta + 2c_3\eta^m(c_1 + c_2)$$

$$B_4 = ((c_2(c_1 - c_2) - 2w(3c_1 - c_2))(c_1 + 2c_2) + 4c_1c_2w)\theta - 2c_1c_3(c_1 + c_2)\eta^m$$

We refer to the numerical simulation for the comparison of profits.

References

Ali AQ, Sultan ABMd, Ghani AAA, Zulzalil H (2019) A Systematic mapping study on the customization solutions of software as a service applications. IEEE Access 7:88196–88217. https://doi.org/10.1109/ACCESS.2019.2925499

Bhargava HK, Choudhary V (2008) Research note—when is versioning optimal for information goods? Manage Sci 54:1029–1035. https://doi.org/10.1287/mnsc.1070.0773

Cheng HK, Li S, Liu Y (2015) Optimal software free trial strategy: limited version, time-locked, or hybrid? Prod OperManag 24:504–517. https://doi.org/10.1111/poms.12248

Choudhary V (2007) Comparison of software quality under perpetual licensing and software as a service. J ManagInfSyst 24:141–165. https://doi.org/10.2753/MIS0742-1222240206

Dutt A, Jain H, Kumar S (2018) Providing software as a service: a design decision(s) model. IseB 16:327–356. https://doi.org/10.1007/s10257-017-0356-9

Fan H, Hussain FK, Younas M, Hussain OK (2015) An integrated personalization framework for SaaS-based cloud services. Future GenerComputSyst 53:157–173. https://doi.org/10.1016/j.future.2015.05.011

Ge C, Huang K-W (2014) Analyzing the economies of scale of software as a service software firms: a stochastic frontier approach. IEEE Trans Eng Manage 61:610–622. https://doi.org/10.1109/TEM.2014.2359975

Gilmore JH, Pine BJ (1997) The four faces of mass customization. Harvard Business Rev 75:91–101

Guo Z, Ma D (2018) A Model of Competition Between Perpetual Software and Software as a Service. MIS Quarterly 42:101–120. https://doi.org/10.25300/MISQ/2018/13640

Jaroucheh Z, Liu X, Smith S (2010) A model-driven approach to flexible multi-level customization of SaaS applications. SEKE 2010 - Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering 241–246.

Jia K, Liao X, Feng J (2018) Selling or leasing? dynamic pricing of software with upgrades. Euro J Oper Res 266:1044–1061. https://doi.org/10.1016/j.ejor.2017.10.063

Katz ML (1984) Firm-specific differentiation and competition among multiproduct firms. J Bus 57:S149–S166. https://doi.org/10.1086/296244

Kong L, Li Q, Zheng X (2010) A Novel Model Supporting Customization Sharing in SaaS Applications. Proceedings 2010 Second International Conference on Multimedia Information Networking and Security (MINES 2010) 225–9 https://doi.org/10.1109/MINES.2010.57

Li J, Liu C, Xiao W (2016) Modularity, lead time and return policy for supply chain in mass customization system. Int J ComputIntelliSyst 9:1133–1153. https://doi.org/10.1080/18756891.2016.1256575

Li S, Cheng HK, Jin Y (2018) Optimal distribution strategy for enterprise software: retail, saas, or dual channel? Prod OperManag 29(11):1928–1939. https://doi.org/10.1111/poms.12909

Ma D, Kauffman RJ (2014) Competition between software-as-a-service vendors. IEEE Trans Eng Manage 61:717–729. https://doi.org/10.1109/TEM.2014.2332633

Ma D, Seidmann A (2015) Analyzing software as a service with per-transaction charges. InfSyst Res 26:360–378. https://doi.org/10.1287/isre.2015.0571

Makki M, Landuyt DV, Lagaisse B, Joosen W (2018) A comparative study of workflow customization strategies: Quality implications for multi-tenant SaaS. J SystSoftw 144:423–438. https://doi.org/10.1016/j.jss.2018.07.014

Moens H, De Turck F (2014) Feature-Based Application Development and Management of Multi-Tenant Applications in Clouds. In: 2014 SPLC 18th International Software Product Line Conference 1: 72–81 https://doi.org/10.1145/2648511.2648519.

Mohamed F, Abu-Matar M, Mizouni R, et al (2014) SaaS Dynamic Evolution Based on Model-Driven Software Product Lines. In: 2014 Ieee 6th International Conference on Cloud Computing Technology and Science (cloudcom). 292–299 https://doi.org/10.1109/CloudCom.2014.131.

Moorthy KS (1984) Market segmentation, self-selection, and product line design. Mark Sci 3:288–307. https://doi.org/10.1287/mksc.3.4.288

Mussa M, Rosen S (1978) Monopoly and product quality. Journal of Economic Theory 18:301–317 https://doi.org/10.1016/0022-0531(78)90085-6

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

Shivendu S, ZhangJames Z (2015) Versioning in the software industry: heterogeneous disutility from underprovisioning of functionality. InfSyst Res 26:731–753. https://doi.org/10.1287/isre.2015.0597

Sun W, Zhang X, Guo CJ, et al (2008) Software as a service: configuration and customization perspectives. 2008 IEEE Congress on Services Part II (SERVICES-2) 18–25 https://doi.org/10.1109/SERVICES-2.2008.29

Thomas P, Fitch A (2019) Oracle to Make 2,000 Hires in Cloud Push. The wall street journal. https://www.wsj.com/articles/oracle-to-make-2-000-hires-in-cloud-push-11570539658. Accessed 20 September 2020

Tsai W-T, Shao Q, Li W (2010) OIC: Ontology-based intelligent customization framework for SaaS. In: 2010 IEEE International Conference on Service-Oriented Computing and Applications (SOCA). 1–8 https://doi.org/10.1109/SOCA.2010.5707139

Vaquero LM, Rodero-Merino L, Caceres J, Lindner M (2009) A break in the clouds: towards a cloud definition. ACM SIGCOMM ComputCommun Rev 39:50–55. https://doi.org/10.1145/1496091.1496100

Varian HR (2000) Buying, sharing and renting information goods. J Ind Econ 48:473–488. https://doi.org/10.1111/1467-6451

Zhang J, Seidmann A (2010) Perpetual versus subscription licensing under quality uncertainty and network externality effects. J ManagInfSyst 27:39–68. https://doi.org/10.2753/MIS0742-1222270103

Ziani D, AlShehri A (2015) A new framework for customizing ERP systems in a multi tenant SaaS environment. In: 2015 2nd World Symposium on Web Applications and Networking (WSWAN) 1–7 https://doi.org/10.1109/WSWAN.2015.7209089.