Economies of Scale in the Software Industry

Working Paper

Fried-Junior SABAYE[1]

November 2025

Abstract

This paper revisits, using internal accounting data, the widely held view that software publishing is characterized by strong increasing returns to scale. We exploit a quarterly dataset spanning 2011–2024 for a French software publisher that successively transformed its production and distribution model: from a traditional *on-premise* offering to *SaaS*, and subsequently to a *Full Web* architecture deployed alongside legacy systems. The objective is to identify how these technological transitions reshape cost structures and alter the dynamics of returns to scale.

Empirically, we estimate Cobb–Douglas and translog cost functions in a setting marked by substantial collinearity across expenditure categories, relying on regularization techniques to stabilize inference. Structural break tests, sub-period estimations, and rolling-window analyses consistently reveal regime shifts that align closely with the firm's technological transformation phases.

The results indicate that returns to scale vary markedly over time. Prior to the *SaaS* phase, the sum of elasticities is close to 0.70. As *SaaS* adoption intensifies, it typically declines toward 0.60, with episodes reaching values as low as 0.25, indicating markedly stronger increasing returns to scale. In the *Full Web* phase, the sum of elasticities exceeds unity (around 1.12), pointing to the emergence of diseconomies of scale associated with integration, coordination, and support constraints. The translog specification further uncovers rising marginal costs (saturation effects) and cross-department complementarities that increase average unit costs through higher coordination requirements. Taken together, these findings provide an empirical, cost-based assessment of how returns to scale evolve throughout the transition from *on-premise* to *SaaS* and ultimately to *Full Web*.

Keywords : Economies of scale • Software industry • Cost functions • Penalized regression • Cost allocation

JEL classification : L86 • D24 • L11 • L25 • C51 • C52

✉ Fried-Junior SABAYE
sfriedjunior@gmail.com

[1] IRJI Laboratory - EA 7496 - University of Tours

1 Introduction

In advanced economies, the software industry has established itself as a core segment of market services, both through its direct contribution to value added and through its pervasive spillovers across user industries. In Europe, Eurostat structural business statistics indicate that "information and communication" services accounted for 6.6% of the Union's market-sector value added in 2022, with programming, consultancy, and related activities representing 51.1% of value added within this aggregate. At the global level, market estimates place annual software spending above USD 1.1 trillion in 2024, within total IT expenditures of roughly USD 5.04 trillion. Medium-term projections point to continued expansion: global IT spending is expected to remain on an upward trajectory and could exceed USD 7 trillion by 2028, suggesting a persistently favorable environment for software publishers (Gartner, 2024, 2025).

From a microeconomic perspective, the software sector is often introduced through a stylized cost asymmetry: high fixed costs of design and development combined with very low, sometimes near-zero, marginal costs of reproduction and distribution. Such a configuration is consistent with increasing returns to scale. In practice, however, the cost structure observed among software publishers is more nuanced and depends critically on the underlying techno-economic configuration. Changes in commercialization and delivery models give rise to distinct regimes. The historical *on-premise* model—local installation and deployment on client machines—has gradually been challenged and partially supplanted by subscription-based offerings, first in the form of *Software as a Service* (*SaaS*), where software is hosted and operated by the provider and delivered remotely over the Internet, and more recently through *Full Web* application architectures. In the latter, all functionalities and data are natively server-side, accessed via a browser, with continuous updates and a high degree of industrialization of execution functions (deployment, monitoring, security, and compliance) on the publisher's side. This shift, reinforced by intensifying competition, reallocates an increasing share of expenditures from upfront investment toward ongoing operations and customer acquisition and retention, thereby reshaping the cost structure.

Indeed, while code replication remains inexpensive, growth increasingly requires recurrent spending on infrastructure, operations, cybersecurity, and service quality, alongside sustained efforts in customer acquisition and retention. Moreover, periods of technological coexistence can generate temporary duplication—maintenance, support, tooling—raise compliance and security requirements, and introduce operational complexity that alters cost behavior. The central issue is therefore not to reaffirm the potential for increasing returns to scale, but to establish empirically the conditions under which they materialize—or erode—as firms move sequentially through *on-premise*, *SaaS*, and *Full Web* regimes. Put differently, the objective is to characterize how the cost function is recomposed during these transitions and to assess whether gains from large-scale diffusion are partially or fully offset by rising operating, coordination, and complexity costs, particularly during phases of regime overlap.

To address this question, the analysis relies on a quarterly time series of 56 observations reconstructed from accounting, managerial, and human-resources systems, allowing a fine-grained view of cost dynamics over time for a mid-sized software publisher. The paper pursues three complementary objectives. First, it estimates cost elasticities associated with major expenditure centers and quantifies returns to scale, explicitly accounting for structural collinearity across cost categories through a regularized log-log (Ridge) specification. Second, it dates regime shifts and distinguishes transitory shocks from persistent reconfigurations of the cost–activity relationship using structural break tests and sub-period estimations aligned with technological milestones. Third, it documents the mechanisms underlying these changes—nonlinearities, coordination costs, and cross-department complementarities—through a guided translog extension estimated via Ridge regression and a rolling-window analysis of returns to scale, with inference robust to temporal dependence. By combining internal managerial accounting, technological segmentation, and regularized econometrics, the study provides an empirical account of how the transition toward operated architectures (*SaaS* and then *Full Web*) reshapes a software publisher's cost function, and to what extent expected returns to scale may first be reinforced and later temporarily neutralized by rising operating and coordination costs.

2 Literature Review

2.1 From coding effort to lifecycle expenditures

The literature internal to software engineering has long approached cost through its most directly observable component: development effort. Effort estimation models such as SLIM (Putnam, 1978) and COCOMO (Boehm, 1981) relate measures of software size—lines of code or function points—to labor input (person-months) in order to produce cost and schedule forecasts useful for project management. Later extensions, most notably COCOMO II, refined adjustment factors and empirical calibration while preserving this operational focus (Boehm et al., 2000). This body of work provides a robust framework for budgeting initial engineering effort, but it offers limited insight into the structure of expenditures once software enters production, diffuses, is commercialized, and must be operated over time.

A second strand of research shifts attention toward the dynamic economics of software projects, emphasizing that costs are not a simple function of code volume. System dynamics models highlight feedback loops among schedules, productivity, quality, and coordination: under time pressure, acceleration and late staffing increases can raise rework and communication costs, leading to total effort exceeding initial plans (Abdel-Hamid & Madnick, 1991). This perspective formalizes the intuition behind Brooks' Law and underscores that cost trajectories depend on organizational mechanisms—planning, learning, quality control—rather than on coding effort alone.

The most relevant extension for the present study concerns the post-deployment phase. As software becomes a durable asset, maintenance and functional evolution account for a substantial share of cumulative expenditure. Using data from industrial teams, Banker and Slaughter (1997) document economies of scale in maintenance: batching change requests into releases can reduce average cost per change by pooling preparation, testing, and coordination, while introducing a trade-off with waiting costs when fixes are deferred. More broadly, classic surveys converge on the conclusion that post-delivery activities represent a major component of lifecycle costs (Pigoski, 1997; Wiederhold, 2006).

Taken together, these contributions progressively shift the object of analysis from "development cost" to "lifecycle cost." Yet even when post-launch expenditures are acknowledged, the literature rarely provides tools to estimate a cost function capable of (i) measuring returns to scale and (ii) identifying regime changes when publishers simultaneously modify technology and commercialization models.

The necessary detour through accounting-based approaches

To move beyond the development–maintenance focus, part of the literature draws on management accounting tools. Activity-Based Costing (ABC) aims to trace costs to the activities that generate them and then allocate them to cost objects (products, projects, customers), thereby improving the functional interpretation of expenditure. Applied to information systems projects, the approach of Ooi, Soh, and Lee (2003) illustrates how fine-grained decomposition—analysis, design, development, testing, deployment, support—can better align estimation, execution, and feedback.

These methods, however, face a key limitation with respect to the present question. On the one hand, they primarily address an allocation issue ("which costs are associated with which activities?") rather than an elasticity issue ("how does total cost respond when activity scales up?"). On the other hand, their implementation depends heavily on internal traceability, and the organizational costs of collecting and maintaining cost drivers can be substantial, motivating more parsimonious variants such as time-driven ABC (Kaplan & Anderson, 2004). In short, ABC and TCO approaches enrich cost description and structure accounting information, but they do not in themselves provide a standard empirical framework for estimating returns to scale or comparing successive techno-economic regimes.

Cloud computing, cost reconfiguration, and the renewed question of scale

The transition to cloud computing strengthens the need for an explicit cost-function framework, as it alters the relationship between scale, architecture, and expenditure. Moving toward infrastructure and platform services transforms some upfront investments into operating expenses sensitive to usage, service levels, and integration complexity. Economically, this shift is not merely an accounting reclassification: it is accompanied by mechanisms that can raise switching costs—portability constraints, interoperability issues, learning effects, and technical dependencies—and lock in customers through pricing and contractual structures. Economic and competition analyses of cloud markets document these issues, including the role of exit or *egress* fees in raising migration costs (Autorité de la concurrence, 2023; CERRE, 2024; OECD, 2025).

These mechanisms have direct implications for returns to scale among software publishers. Even if code replication remains inexpensive, growth may require recurrent spending on availability, monitoring, security, compliance, and operations, whose curvature depends on technical choices (multi-tenancy, redundancy, observability), organizational arrangements (operations/SRE, support), and commercial commitments (SLAs, product segmentation). Strong increasing returns to scale thus remain a plausible intuition, but they become an empirical hypothesis to be tested regime by regime rather than a guaranteed structural property.

In a similar vein, cloud-native and *serverless* architectures make cost models more explicitly usage-based. Jonas et al. (2019) synthesize the economic characteristics of *serverless* computing—on-demand execution, auto-scaling, fine-grained billing—while Adzic and Chatley (2017) report cost reductions following migration to AWS Lambda, attributed to reduced overprovisioning and closer alignment between capacity and demand. For cost analysis, the point is not to generalize these findings mechanically, but to recognize that cost structures become more sensitive to architecture, service levels, and demand variability, reinforcing the relevance of explicitly estimating cost functions.

2.2 Measuring output in software services

Estimating a cost function in services first raises the issue of output definition. Unlike goods industries, where standardized physical units are available, services combine heterogeneity, co-production with customers, and quality differentiation. Software compounds these challenges: diverse product lines and modules, coexistence of licenses and subscriptions, heterogeneous pricing, and the provision of a continuous service that includes maintenance, support, and updates. In this context, searching for a single "natural" output measure is rarely realistic; a more robust approach consists in constructing an empirical compromise that balances accounting observability, economic relevance, and statistical feasibility.

Applied service-sector studies reveal a common pattern: when a natural unit of output is not directly observable, activity proxies are used, sometimes complemented by structural or quality variables. Depending on the setting, these proxies may be physical (units served), monetary (value captured), or mixed (volume and value) when production is multi-product and prices vary across the bundle. Illustrative examples abound. In cultural services such as museums, attendance can serve as a synthetic indicator of production scale, augmented with variables reflecting service heterogeneity and quality (Jackson, 1988). In air transport, output may be represented by composite units (passenger-kilometers, ton-kilometers) and enriched with network or operational attributes when aggregation masks technological or organizational differences (Gillen, Oum, & Tretheway, 1990). In telecommunications, combining a base indicator (subscribers) with a value indicator (revenues) allows analysts to capture both activity volume and value in multi-service environments (Bagadeem, 2021). In port infrastructure, production is often measured through multiple physical traffics, supplemented by proxies when certain service dimensions are not directly quantifiable (Jara-Díaz et al., 2002). These cases do not exhaust possible approaches; rather, they illustrate a general empirical principle: output measures are constructed to be measurable, economically interpretable, and compatible with estimation constraints.

Output choices are inseparable from input specification. In the standard microeconomic framework, the cost function relates minimum expenditure to output levels and a vector of factor prices, enabling analysis of substitution and theoretical properties when data permit (Shephard, 1953). In service-sector applications, however, inputs vary with data availability and analytical objectives. A recurring trade-off emerges between, on the one hand, a “dual” representation close to theory (prices of labor, capital, and intermediates) and, on the other, more operational variables when factor prices are difficult to observe or when the goal is to identify organizational cost drivers. Again, applied studies are instructive. In services such as museums or port infrastructure (Jackson, 1988; Jara-Díaz et al., 2002), analysts typically (a) include a core set of productive inputs via factor prices when feasible; (b) otherwise rely on coherent accounting proxies (wage bills, operating expenses, capital costs); and (c) add structure or quality variables to mitigate omitted-variable bias. Bagadeem (2021), for instance, uses directly observable cost categories (capital, marketing and sales, other operating expenditures), while Gillen, Oum, and Tretheway (1990) aggregate inputs into blocks (labor, capital, operating inputs) and associate each with a unit price consistent with observed accounts and volumes, allowing estimation of a multi-product cost function without excessive parameter proliferation.

These regularities motivate the choices adopted in this paper. On the output side, a dual measure—volume and value—captures both operational scale and value captured (price effects, upmarket moves, mix changes) without multiplying output dimensions to the point of undermining estimation on short time series. On the input side, the objective is not to reconstruct a “pure” technology based on exhaustive factor prices—often unavailable and heterogeneous in internal data—but to identify how internal resource reallocation reshapes the cost function across commercialization regimes. In this perspective, decomposing expenditures by operational departments provides a directly observable and managerially meaningful proxy for functional intensity—R&D, marketing, sales, support, and others—consistent with how firms actually arbitrate resources. Issues related to functional specification and the treatment of collinearity are addressed in the next section.

2.3 Functional forms and identification

The Cobb–Douglas specification in logarithms provides a natural starting point, as it yields directly interpretable elasticities and allows an immediate discussion of returns to scale. In our setting, we adopt a log-log specification in which average unit cost is explained by outputs (volume and value) and by inputs constructed from a decomposition of expenditures across operational departments. The baseline form can be written as follows:

$$\ln C_t = \ln \alpha_0 + \gamma_1 \ln units_t + \gamma_2 \ln revenue_t + \sum_{i \in D} \alpha_i \ln(D_{i,t}) + \varepsilon_t$$

where (C_t) denotes average unit cost, $units_t$ the volume measure, *revenuet* the value measure, and Di, t the variables associated with operational departments: R&D, marketing, the sales force, technical support, and other functions. The intercept (α_0) should not be overinterpreted as a "pure fixed cost," although it absorbs the residual component not captured by the explicit inputs and by the allocation conventions used.

The translog specification (Christensen, Jorgenson & Lau, 1973) extends this framework by allowing curvature and interactions across variables, which is particularly appropriate when complementarities (or tensions) across departments and nonlinear effects related to technological transitions are plausible:

$$\begin{aligned}\ln C =& \alpha_0 + \sum_{k=1}^{2} \gamma_k \ln Y_k + \sum_{i=1}^{5} \alpha_i \ln D_i + \tfrac{1}{2} \sum_{k=1}^{2} \sum_{\ell=1}^{2} \delta_{k\ell} \ln Y_k \ln Y_\ell \\ &+ \tfrac{1}{2} \sum_{i=1}^{5} \sum_{j=1}^{5} \beta_{ij} \ln D_i \ln D_j + \sum_{i=1}^{5} \sum_{k=1}^{2} \theta_{ik} \ln D_i \ln Y_k + \varepsilon.\end{aligned}$$

where $Y_{1,t}$ and $Y_{2,t}$ are the two output measures (revenue and units), and $\mathcal{D}$ denotes the set of departments. The usual symmetry restrictions apply $(\delta_{k\ell} = \delta_{\ell k})$, $(\beta_{ij} = \beta_{ji})$. The cost of this flexibility is a rapid increase in the number of parameters, which becomes especially problematic when the time series is short and cost categories move jointly.

This difficulty is well documented in the applied cost-function literature: the translog is appealing because it accommodates curvature and interactions, but it tends to amplify multicollinearity, especially once quadratic and cross terms are introduced. The standard responses combine several tools: theoretical restrictions (symmetry, homogeneity, and sometimes concavity); system estimation with share equations derived from Shephard's lemma when price/quantity data are available; normalization and centering around an expansion point to improve numerical stability; or, more pragmatically, simplifying the functional form when the sample is limited, since instability of second-order effects is a recurrent empirical finding in some applications (Gillen, Oum & Tretheway, 1990). In our setting, these options are partly constrained by the nature of the data (internal series, short sample, and strong structural correlations across departments). This creates a methodological requirement: retain economically central variables while stabilizing inference.

This is precisely where the paper links functional forms and regularization in an explicit way, by clearly separating their roles. The Cobb–Douglas specification provides an interpretable baseline; it is estimated under Ridge regularization and serves as the main model for recovering stable elasticities while retaining the full set of departments. The translog is then introduced as a flexible extension—not to replace the regularized Cobb–Douglas, but to deepen the analysis of returns to scale by allowing nonlinearities and interactions. Estimating it requires a more selective regularization strategy to control the explosion in parameters; we therefore rely on manual selection.

2.4 Penalized regressions as an empirical remedy for multicollinearity

A software publisher's income statement almost mechanically generates strong correlations across expenditure categories. The development and release of a new product, for instance, typically coincide with an upstream intensification of R&D and, at launch, a surge in marketing effort (promotion, campaigns, go-to-market), a reinforcement of support and customer assistance (more tickets, onboarding and customer assistance), and, downstream, a rise in operating costs (capacity, monitoring, security, and compliance). In addition, transitions across regimes (*Desk*, *SaaS*, *Full Web*) often involve overlap phases in which multiple technical and commercial chains coexist. These co-movements make inputs collinear: variance inflation factors (VIFs) can become large, and small changes in the estimation window may flip the sign of elasticities that are economically central. In such conditions, unpenalized estimation—whether an enriched Cobb–Douglas or a translog—tends to deliver unstable coefficients: fit may remain excellent, but inference and interpretation become fragile, especially for interaction terms.

Statistical penalization provides a natural response to this collinearity problem. Rather than dropping variables through stepwise procedures, regularization retains the information set while constraining coefficient magnitudes, preventing the familiar "blow-up" of estimators when regressors are redundant. Intuitively, it downweights purely redundant variation carried by highly correlated variables and emphasizes the component that is genuinely informative for explaining cost, thereby improving out-of-sample stability and the economic coherence of estimated elasticities.

In this setting, Ridge regression (Hoerl & Kennard, 1970) is the most appropriate tool when the goal is to retain all departments—exactly the case here, since the purpose is to track how cost hierarchies and functional elasticities evolve without arbitrarily eliminating an operational dimension. Ridge amounts to solving:

$$\min_{\beta} \left\|\mathbf{y} - \mathbf{X}\beta\right\|_2^2 \; + \; \lambda\|\beta\|_2^2,$$

where λ controls shrinkage. The (L2) penalty corresponds to Tikhonov regularization: it strongly stabilizes estimates under multicollinearity, keeps all variables in the model, and reduces the influence of structural redundancies. In the paper, the Ridge-estimated Cobb–Douglas therefore plays the role of the reference specification: it delivers stable, comparable, and directly interpretable elasticities.

The LASSO (Tibshirani, 1996), based on the (L1) penalty, is defined as:

$$\min_{\beta} \left\|\mathbf{y} - \mathbf{X}\beta\right\|_2^2 \; + \; \lambda\|\beta\|_1$$

which encourages exact zeros and produces automatic sparsity. This property is useful for exploring redundancy across variables, but it can be problematic in a department-level cost function: when several departments are strongly correlated, the LASSO may select one and drop another, at the risk of removing an operational dimension that is economically essential. In our setting, it is therefore mainly used as a comparison benchmark and diagnostic tool, but it cannot serve as the main estimator if the goal is to preserve the full set of departments.

Elastic Net (Zou & Hastie, 2005) combines the (L2) and (L1) penalties:

$$\min_{\beta} \left\|\mathbf{y} - \mathbf{X}\beta\right\|_2^2 + \lambda\Big[(1-\alpha)\|\beta\|_2^2 + \alpha\|\beta\|_1\Big].$$

In our setting, however, Elastic Net provides only marginal gains. It is mainly useful when one simultaneously wants to stabilize a very rich specification and automatically select a subset of variables from a large pool of correlated candidates. Here, by contrast, the paper's purpose is precisely to retain all cost inputs (and, in the translog extension, the small set of second-order terms chosen *ex ante*) in order to compare successive techno-economic regimes without having the presence or absence of a term depend on an algorithmic choice. In a highly correlated environment, the L1 component of Elastic Net would tend to set some coefficients or interactions to zero via “arbitrage” among redundant regressors, making inter-period comparisons less transparent (a term may disappear in one phase and reappear in another). Put differently, once the specification is guided and the priority is stabilization rather than sparsity, Ridge is sufficient: it addresses collinearity while preserving the full set of dimensions required for economic interpretation and phase comparisons.

2.5 Summary and positioning

The software-focused literature offers robust tools to budget development effort and to explain certain project and maintenance dynamics. It provides much less guidance, however, for estimating—on longitudinal internal accounting data—a cost function capable of measuring returns to scale and identifying breaks associated with technological and commercial transitions. Accounting approaches, for their part, describe expenditure composition in detail but only imperfectly answer the question “how does cost respond to growth and regime change?” Finally, the economics of services provides methodological benchmarks for output measurement and motivates flexible functional forms, provided that collinearity and small-sample constraints are handled appropriately.

This paper sits at that intersection. By combining a dual activity measure based on volume and value, a department-level decomposition of expenditures, and regularized estimation, the study aims to characterize empirically how a publisher's returns to scale evolve as its model is reshaped across technological regimes. The comparison across specifications (standard and penalized Cobb–Douglas, then penalized translog) is not intended to replace one functional form with another, but to build a progressive reading: first stabilize functional elasticities in a model that retains all departments, then, in a second step, explore nonlinearities and interactions that may illuminate the dynamics of returns to scale during the transitions *Desk* $\rightarrow$ *SaaS* $\rightarrow$ *Full Web*.

3 Data and variable construction

The study relies on a quarterly time series of fifty-six observations covering 2011-T1 to 2024-T4. The firm employs more than 500 workers and generates over €50 million in revenue by the end of the period. All data come from the accounting and management-control systems of the firm under study—a French mid-sized publisher of management software—ensuring consistency between the financial statements produced and the analytical dataset. Each quarterly observation aggregates, on the one hand, general-ledger entries (i.e., all direct and indirect expenses recorded in financial accounting) and, on the other hand, managerial accounting entries that allocate these expenses across functional departments.

Payroll records and HR histories are matched to these financial data. They allow the reconstruction, for each department, of headcounts, wage bills, and potential perimeter changes—inputs needed to ensure a consistent allocation of expenses across operational departments over the full period.

For predictive evaluation, the dataset is split into two subsamples. The fifty-two quarters from 2011-T1 to 2023-T4 constitute the estimation sample. The four quarters of 2024, excluded from calibration, form an independent test sample used to assess out-of-sample accuracy (MAPE, RMSE, and a synthetic overall-accuracy indicator).

The depth of the series, combined with the fine disaggregation of accounting items and the stability of collection procedures over more than a decade, provides an empirical foundation well suited to comparing functional forms and estimation methods in the sections that follow.

Data extractions were conducted under a non-disclosure agreement (NDA). Quarterly aggregates are anonymized and scaled for publication; no customer identifiers or individual compensation information are included.

3.1 Variables and scope

As discussed above, designing a cost function adapted to the software sector is challenged both by the diversity of expenditure categories and by the plurality of production modes. To better reflect this reality, we propose a generalized model in which the dependent variable is average unit cost—total cost divided by the number of units sold.

In practice, a simple count of sales (*'units'*) does not reflect product heterogeneity: an add-on may sell for a few euros, whereas a mid-market software package can be priced in the thousands. Conversely, revenue (*'revenue'*) captures value dispersion but obscures purely quantitative dynamics. By combining these two indicators, we capture the two fundamental dimensions of activity.

On the **input** side, we use a departmental decomposition, each category being normalized by the number of units sold: R&D (*RnD*), Marketing (*marketing*), the sales force (*salesforce*), Technical Support (*'AT'*), and Other functions (HR, general secretariat, top management, business units grouped under *Others*). For each department, we include wages, overhead and miscellaneous expenses per employee, and subcontracting costs specific to the department. This level of granularity follows management-control practice, where operating costs are reallocated according to each department's share of headcount, yielding a more detailed mapping between a unit sold and the resources mobilized to deliver it. The precise mapping of accounts to departments, as well as the allocation rules for common expenses (rent, cloud, miscellaneous expenses), are reported in Appendix A to facilitate replicability.

All variables are expressed in natural logarithms. This transformation ensures (i) zero-degree homogeneity of the function—each coefficient is interpreted as a constant elasticity—(ii) reduced skewness of cost distributions, and (iii) automatic absorption of level effects. All cost components are also divided by the number of units sold, so that they measure an average cost per unit sold: the intercept then collects only the fixed and variable fragments not attributable to a department or to employees, namely SaaS costs (servers, hosting, bandwidth), royalties, and taxes.

3.2 Stationarity tests

Before estimating dynamic models, it is necessary to verify that the time series do not contain unit roots: without stationarity, estimated relationships may be spurious and asymptotic statistics (t, F tests, etc.) become invalid. We therefore apply three complementary procedures: the Augmented Dickey–Fuller test (Dickey & Fuller 1979), the Phillips–Perron test (Phillips & Perron 1988), and the KPSS test, which instead takes stationarity as the null hypothesis (Kwiatkowski et al. 1992). Detailed results reported in Appendix B show that, after log transformation and first differencing, all series are integrated of order zero. We can therefore estimate the models presented below (log-log Cobb–Douglas, Ridge and Lasso regularizations) using these variables without concern for residual unit roots.

4 Estimation and method selection

This section develops a generalized model to analyze the cost structure of a software publisher across its operational departments, test whether returns to scale are increasing or decreasing, and produce reliable cost predictions. Three econometric approaches are compared: a Cobb–Douglas specification and two regularized variants, Ridge and Lasso. Economically, the key questions are (1) the sum of coefficients (excluding the intercept) as a measure of returns to scale, (2) the relative weight of each department in the cost structure, and (3) predictive performance on the test period.

Table 1 reports estimated coefficients over the 2011–2023 training sample: on the left, results from the Cobb–Douglas model with significance levels; on the right, coefficients obtained when the same specification is estimated under Ridge and LASSO penalization, consistent with the methods defined above. This layout makes it possible to compare, holding variables fixed, how sensitive the elasticities are to regularization.

Table 1: Estimation on the training data

Cobb-Douglas				Penalized regressions	
Variables	Coeff	Signif.		Ridge	Lasso
(Intercept)	1.041	.		2.752	1.748
revenue	-0.0163			0.001	-0.008
units	0.058			-0.074	0
RnD	0.427	***		0.383	0.423
marketing	0.089	***		0.112	0.089
salesforce	0.237	***		0.107	0.186
Others	0.137	***		0.126	0.131
tech_support	0.159	***		0.161	0.159
ΣB	1.091			0.816	0.98
Explained var.	99.26			98.96	99.32
			RMSE	0.022	0.113

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

The Cobb–Douglas model delivers an extremely high share of explained variance (above 99%), which is unsurprising in our setting: the equation links average unit cost to its internal components (expense items allocated across departments) and to two activity measures. The goal is to quantify the relative weight of each component and track how these weights shift across techno-economic regimes. By contrast, inference on the output variables is fragile: the coefficients on *units* and *revenue* are not statistically significant, providing an early indication of multicollinearity-driven instability.

Table 2: VIF for the Cobb–Douglas model

revenue	units	RnD	marketing	salesforce	Others	tech_support
4.4	47.8	3.01	2.45	44.25	2.03	3.12

This instability is confirmed by *Table 2*: variance inflation factors are particularly high for *units* and *salesforce* (VIFs above 40), far beyond conventional thresholds (typically between 5 and 10; Gujarati & Porter, 2009), indicating substantial redundancy among regressors. Economically, this correlation is expected: quantities sold and selling expenditures—especially when part of variable compensation is tied to targets—tend to move together.

Ridge introduces an (L2) penalty that shrinks coefficients without setting any of them to zero. Its key advantage here is precisely to stabilize inference under collinearity while retaining both outputs and all departments. The results show that penalization primarily corrects the collinearity hotspot identified by the VIFs: the *salesforce–units* pair is sharply rebalanced, with a pronounced drop in the *salesforce* coefficient (from 0.237 to 0.107) and a sign reversal for *units*, which becomes negative (from 0.058 to -0.074). This restores a pattern more consistent with the standard economies-of-scale intuition, whereby higher volume dilutes unit costs. By contrast, several departments remain of similar magnitude (*Marketing*, *AT*, *Others*), while *RnD* is slightly reduced. In other words, Ridge mainly operates where statistical redundancy is most severe and where OLS was least stable, yielding more robust elasticities (e.g., *RnD* falls from 0.427 to 0.383). The sum of elasticities ($\Sigma_\beta \approx 0.82$) points to an overall regime of decreasing returns in this collinearity-adjusted specification, while the in-sample fit remains tight (RMSE = 0.022).

LASSO applies an (L1) penalty that can set coefficients exactly to zero and automatically select a subset of variables. The coefficients on retained variables are broadly consistent with the Ridge structure, but the procedure sets *units* to zero. In our framework, keeping both output dimensions (volume and value) is essential: dropping *units* removes a core channel through which scale effects operate and makes the economic interpretation incomplete. For this reason, LASSO is used as a benchmark and diagnostic for potential redundancies, but it is not retained for the main analysis, which requires the full set of variables to be preserved throughout.

4.1 Out-of-sample performance

Predictive robustness is assessed on the four quarters of 2024, which are held entirely out of estimation. We use two complementary metrics: (i) MAPE, which measures mean percentage error and captures the "relative" accuracy of forecasts, and (ii) total bias, which compares the sum of predicted values to the sum of observed values and detects systematic drift (over- or under-prediction).

The first metric is MAPE (Mean Absolute Percentage Error), defined as the average absolute percentage deviation between predicted values ($\widehat{y}_i$) and realized values (y_i):

$$\text{MAPE} \;=\; \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\widehat{y}_i - y_i}{y_i}\right| \times 100.$$

A lower MAPE indicates a stronger ability to forecast the order of magnitude of costs.

The second metric is total bias, defined as the relative gap between the sum of predictions and the sum observed:

$$\text{Aggregate}_{\text{bias}} = \frac{\sum_i \hat{y}_i - \sum_i y_i}{\sum_i y_i} \times 100.$$

A negative value indicates overall underestimation, a positive value overestimation; the closer the metric is to zero in absolute value, the better calibrated the forecast.

All three models achieve small errors on this test set (*Table 3*). Cobb–Douglas yields the lowest MAPE (1.79%) and near-zero bias (-0.25%), but this result should be interpreted cautiously: the unregularized specification is precisely the one most exposed to multicollinearity, and its elasticities can flip signs when the training window changes. LASSO also performs well (MAPE = 1.86%), but at the cost of excluding *units*; since our framework requires preserving both volume and value, this model is not retained despite its apparent accuracy.

Table 3: Performance on the test data

Model	MAPE	Aggregate_bias
Cobb-Douglas	1.79	-0.25
Ridge	2.40	-1.70
Lasso	1.86	-0.59

In this context, Ridge is validated as the reference specification: its out-of-sample performance remains of the same order of magnitude (MAPE = 2.40%; bias = -1.70%) while satisfying the core constraint of the analysis—retaining all variables and stabilizing elasticities in a highly collinear environment. Put differently, Ridge accepts a small short-horizon accuracy cost in exchange for greater structural robustness and interpretability, which is critical for comparing technological regimes in the remainder of the paper.

4.2 Statistical validation of the retained model and inference

Having selected Ridge as the reference specification based on elasticity stability and out-of-sample performance (*Table 3*), we assess the validity of inference before using the model for dynamic analysis. Two standard residual diagnostics are applied.

First, the Breusch–Pagan test (1979) finds no evidence of heteroskedasticity: the statistic $\chi^2_7 = 5.21$ ($p = 0.63$) does not reject the null of constant variance. Error dispersion therefore does not systematically vary with the activity level or with the predictors. Second, the Ljung–Box test (1978), computed up to lag 4 (one year at quarterly frequency), strongly rejects the null of no autocorrelation: $Q = 18.11$ with $df = 4$ ($p = 0.0012$). Residuals thus exhibit time dependence not captured by the static specification.

Practically, this diagnosis implies separating estimation from inference. Ridge coefficient estimates remain well behaved, but conventional (OLS-style) standard errors are unreliable under autocorrelation. We therefore report, for each coefficient, Newey–West HAC robust standard errors with truncation set to four quarters (lag=4)—a natural choice for quarterly data (annual horizon) and consistent with the lag-selection rule proposed by Newey and West (1994), $L_{NW} = 4, (N/100)^{2/9}$.

Finally, while Ridge provides a robust economic reading when estimated over the full period, a "global" estimate may smooth over regime shifts: the relationship between unit cost, activity, and departmental expenditures may change along technological transitions. To probe the robustness of our conclusions, the remainder of the paper combines three complementary approaches: formal break detection, re-estimation on homogeneous subperiods, and a rolling-window tracking of returns to scale. Throughout, test inference is also based on HAC robust variances; we adopt a 5% decision threshold for the Chow test and a stricter 1% threshold for the global Wald test, which is more demanding when multiple restrictions are tested jointly.

5 Phase-based estimation

5.1 Structural break tests

Chow test

For a candidate break date $t^\star$, the idea is to compare the fit of a model estimated on the full sample with the fit obtained when allowing two distinct parameter vectors on either side of $t^\star$. Concretely, we estimate (i) the model over the entire period $t = 1, \dots, T$, yielding $\mathrm{RSS}_{\mathrm{glob}}$, and (ii) two separate regressions over the subperiods ($[1, t^\star]$) and $[t^\star + 1, T]$, yielding RSS_1 and RSS_2. The Chow statistic is then:

$$F_{\mathrm{Chow}} = \frac{(\mathrm{RSS}_{\mathrm{glob}} - \mathrm{RSS}_1 - \mathrm{RSS}_2)/(k+1)}{(\mathrm{RSS}_1 + \mathrm{RSS}_2)\big/(n_1 + n_2 - 2(k+1))}$$

It follows an $F_{,k+1,;n_1+n_2-2(k+1)}$ distribution under the null hypothesis of parameter stability (Chow, 1960), where $k+1$ is the total number of estimated coefficients, including the intercept. We apply the test to every quarter between 2013-T1 and 2022-T3; the segments 2011-T1 $\rightarrow$ 2012-T4 and 2022-T4 $\rightarrow$ 2024-T4 serve as window margins. p-values are computed using HAC variance (lag = 4) to account for the autocorrelation diagnostics reported above.

Wald test

The Wald test evaluates a set of parametric restrictions under the null hypothesis (H_0) by comparing the unrestricted estimate to an imposed value (or, here, to equality restrictions before/after the break). It therefore allows **joint** testing of multiple restrictions, whereas univariate tests would focus on a single coefficient. The statistic is:

$$F_{\mathrm{Wald}} = (\hat{\beta} - \beta_0)^\top [\mathrm{Var}_{\mathrm{HAC}}(\hat{\beta})]^{-1} (\hat{\beta} - \beta_0)$$

and it is asymptotically distributed as $\chi^2 * q$ (or $F*, q, ; n-q$) under H_0, where q is the number of restrictions tested (Wald, 1943). This test assesses the full set of "before/after" equalities jointly, whereas a t-test would only test one coefficient at a time.

Plotting the F_{Chow} statistic across candidate break dates (Figure 1) provides a compact view of parameter stability over time. Between 2013 and the first half of 2015, the statistic remains low and nearly flat, consistent with a relatively stable regime. From 2015 onward, it crosses a first threshold and then rises in steps, reaching a pronounced peak around 2020-T1. This pattern points less to an instantaneous break than to a **gradual transition**, during which the relationship between unit cost, outputs, and departmental expenditures is progressively reshaped before tipping into a new regime.

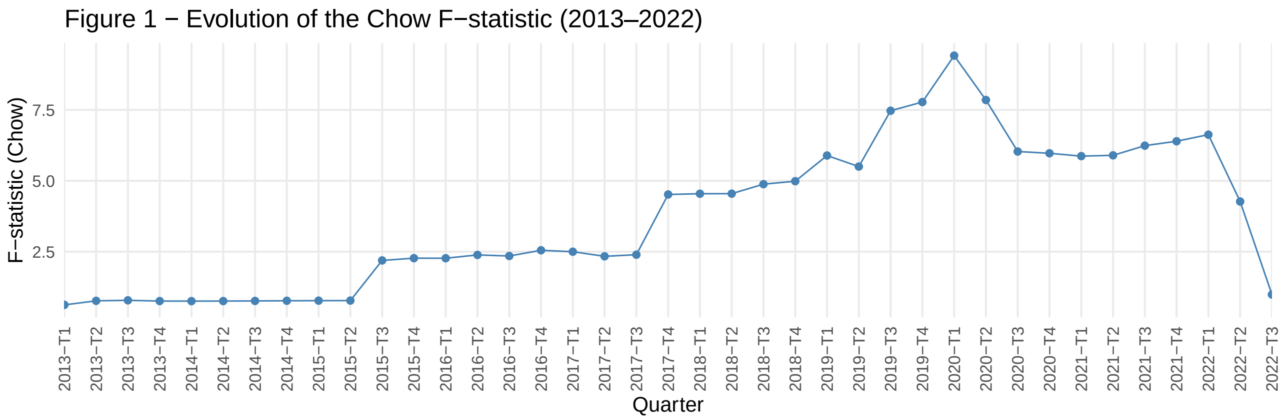


Table 4 then formalizes the decision for the two retained candidate dates, which define the subperiod boundaries used in the rest of the paper. The **2015-T3** point emerges as a plausible regime boundary: the Chow test is right at the 5% threshold ($F = 2.19, p = 0.04899$), while the HAC-robust Wald test (Newey–West, lag=4) clearly confirms a break ($F = 15.17, p \simeq 0$). This gap is consistent with the earlier diagnostics: when residuals are autocorrelated, non-robust inference can become unreliable, which is why we treat the HAC Wald test as the decisive criterion. Economically, this dating is also consistent with *Table 9* (Appendix C), which tracks the firm's activity through the composition of revenue and the allocation of the R&D budget across distribution modes: from 2015 onward, the SaaS share stops being marginal and becomes large enough to alter operating and go-to-market trade-offs (support, operations, release cycle, acquisition and retention)—exactly the type of reconfiguration that can shift the cost function.

Table 4: Break tests at sub-period boundaries (Chow) and HAC-robust tests (Wald–Newey–West)

Period	Chow F	Chow p	Wald F	Wald p	Decision (Wald)
2015-T3	2.1904	0.0489900	15.1762	0	Break confirmed
2020-T1	9.4142	0.0000003	17.1385	0	Break confirmed

The clearest break occurs in 2020-T1. It corresponds to the maximum in Figure 1, and both tests strongly agree (*Table 4*: $F_{\text{Chow}} = 9.41$, $p = 3 \times 10^{-7}$; HAC Wald $F = 18.21, p \simeq 0$). Beyond statistical significance, the shape of the curve is informative: after 2020-T1, F_{Chow} remains elevated for several quarters, pointing to a persistent regime change rather than a purely transitory shock. Appendix C again supports the coherence of this split: the period starting in 2020 coincides with the launch and ramp-up of the *Full Web* program, captured by a rapid increase in the share of the R&D budget allocated to it, implying a durable reallocation of engineering and industrialization expenditures.

The curve stays high until roughly 2022 and then declines, but this movement does not, as it stands, provide a robust dating of a fourth regime. Near the end of the tested window, break statistics become more sensitive to edge effects (shorter post-break subsamples, less information after the tested point), which mechanically compresses the statistic. The post-2022-T1 decline is therefore consistent with relative stabilization within the post-2020 regime combined with sample limitations, rather than with a clear identification of an additional discrete break. Appendix C in fact suggests continuity over these years, marked by the ongoing expansion of SaaS and the gradual intensification of the *Full Web* effort—more akin to a ramp-up than to another sharp structural shift.

These results justify the phase split used for the remainder of the analysis: a *Desk* period (2011-T1 → 2015-T2), a *Desk + SaaS* phase (2015-T3 → 2019-T4), and a *Desk + SaaS + Full Web* phase (2020-T1 → 2024-T4). Figure 1 provides a global map of instabilities, while *Table 4* formally confirms—using HAC-robust inference consistent with residual dynamics—the significance of the two retained boundaries.

5.2 Subperiod estimation

Following the break dating (Figure 1 and *Table 4*), we re-estimate the model separately over each of the three phases. *Table 5* reports, for *Desk*, *Desk + SaaS*, and *Desk + SaaS + Full Web*, the Ridge coefficients together with Newey–West HAC robust standard errors (lag=4), presented as indicative inference, as well as summary metrics. Because each subperiod contains only 18–20 quarters and serial dependence persists, inference is complemented with a moving-block bootstrap: 1,000 replications built from contiguous four-quarter blocks, in line with recommendations for short time series (Künsch, 1989; Lahiri, 2003). The bootstrap results (Appendix D) confirm the orders of magnitude highlighted by HAC-based inference.

Table 5: Ridge estimates with HAC standard errors and summary statistics by subperiod

	Desk		Desk + SaaS		Desk + SaaS + Full Web	
	Coef	SE_HAC	Coef	SE_HAC	Coef	SE_HAC
(Intercept)	3.551	0.245	4.029	1.270	1.708	1.444
revenue	0.003	0.002	-0.001	0.008	-0.022	0.038
units	-0.136	0.016	-0.154	0.087	0.006	0.147
RnD	0.205	0.025	0.267	0.015	0.463	0.022
marketing	0.111	0.007	0.133	0.010	0.118	0.012
salesforce	0.232	0.023	0.139	0.080	0.177	0.061
Others	0.141	0.009	0.048	0.005	0.147	0.023
tech_support	0.157	0.006	0.195	0.013	0.151	0.078
ΣB	0.712		0.626		1.039	
%Var	99.80		99.68		98.32	
N	18		18		20	

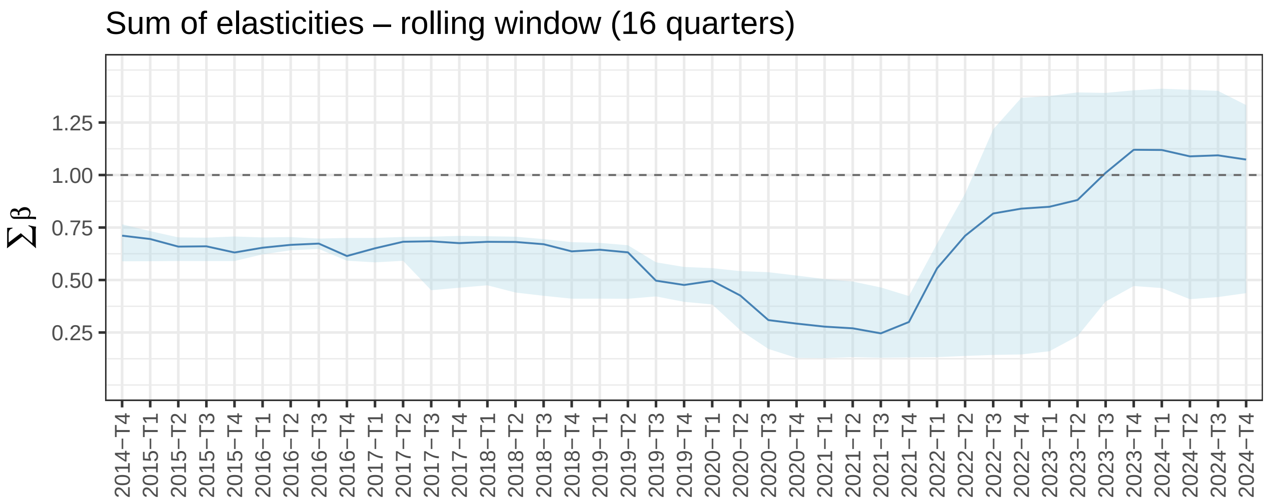
Two findings structure the interpretation. First, Σ_β implies markedly different returns to scale across phases: it remains well below one in *Desk* (0.712) and in *Desk + SaaS* (0.626), then crosses above one in *Desk + SaaS + Full Web* (1.039). In other words, the pre-2020 pattern is consistent with unit costs being diluted as activity expands, whereas after 2020 the marginal dynamics become globally more expensive. Second, the shift is driven by changes in the departmental hierarchy: the *R&D* elasticity rises monotonically (0.205 → 0.267 → 0.464), pointing to a durable reorientation of unit costs toward technical effort and industrialization; *salesforce* declines in the *Desk + SaaS* phase (0.232 → 0.139) and then rebounds in the *Full Web* phase (0.179), consistent with a more intensive migration/relaunch period; the *Others* category contracts sharply in *Desk + SaaS* (0.141 → 0.048) and then increases again (0.146), suggesting a reconfiguration of peripheral overhead around the new regime.

The output coefficients are also highly informative: *units* is negative on average in the first two phases (−0.136; −0.154), then becomes close to zero in the *Full Web* phase (0.006)—with substantial uncertainty ($SE_{HAC} = 0.147$)—which is consistent with a fading "volume dilution" effect once operating and complexity costs dominate. The coefficient on *revenue* becomes more strongly negative in the *Full Web* phase (−0.022), suggesting that, in this regime, the *value* channel has taken precedence over the *volume* channel in the dilution of unit costs.

These results motivate (i) a closer look at the time path of returns to scale and (ii) a complementary, more flexible specification to probe the role of nonlinearities and interactions.

5.3 Time-varying economies of scale

Rather than restricting attention to a static phase comparison, we track the trajectory of returns to scale over time using a rolling estimation. For each quarter (t), the log-log Ridge model is re-estimated on a 16-quarter (four-year) moving window ending in t. This window length provides enough observations to select λ by cross-validation while smoothing short-run fluctuations. Uncertainty around Σ_β is assessed via a **moving-block bootstrap** (1,000 replications, four-quarter blocks): for each window we obtain the empirical distribution of Σ_β and derive a 95% confidence interval.



Between 2014-T4 and 2021-T3, the rolling sum of elasticities, Σ_β, stays well below one. The firm therefore operates under pronounced increasing returns to scale, which become stronger as SaaS gradually ramps up. The trough—around 0.25 between 2020-T3 and 2021-T3—marks the point at which activity growth, particularly through subscriptions, maximizes the dilution of internal cost components in average unit cost. This episode coincides with the Covid-19 shock and the widespread shift to remote work, which boosted demand for cloud solutions and amplified short-run volume effects, while improving the ability to spread fixed operating, support, and go-to-market costs over a rapidly expanding customer base.

From 2022-T1 onward, the pattern reverses. Σ_β rises quickly (back to roughly 0.80 within a year) and crosses the threshold of one from 2023-T3. In this region, the confidence band widens mechanically because the 16-quarter rolling window still mixes late observations from the *Desk + SaaS* regime with the first quarters associated with *Full Web*. Even so, the signal is clear: beyond 2023-T3, the 95% interval lies entirely above one. This marks entry into a third regime characterized by diseconomies of scale, in which engineering (R&D), operations, and support costs—together with cloud run costs—grow more than proportionally with activity.

By late 2024, Σ_β edges down slightly toward 1.05 but does not fall back below one, so the firm does not return to the strong increasing returns to scale observed earlier. Overall, the trajectory is unambiguous: increasing returns to scale strengthen during the *Desk* $\rightarrow$ *SaaS* transition—temporarily boosted during the Covid period— then move into decreasing returns to scale once *Full Web* becomes structural and the coexistence of multiple technical and commercial stacks durably raises the effective marginal cost.

6 Flexible extension

6.1 Translog specification and identification constraints

The penalized Cobb–Douglas is our anchor: it delivers a stable measure of average elasticities and returns to scale, while keeping the interpretation parsimonious and comparable across sub-periods. To deepen this first-pass reading without altering the empirical setup (same outputs, same cost centers, same segmentation), we then introduce a **guided** translog extension. The aim is strictly diagnostic: (i) test for curvature in R&D effort, and (ii) explore a small set of organizational interactions whose interpretation is economically meaningful in a software publisher's operating model (complementarities or frictions across functions). In that sense, the translog is a pragmatic tool designed to describe how the *SaaS* and then *Full Web* transitions reshape the cost structure through internal interdependencies and potential saturation effects.

Given the limited size of the subsamples (18 to 20 quarters), the translog specification is deliberately restricted to the most informative and interpretable second-order terms. We first include a quadratic term, $R\&D^2$, to capture potential concavity/convexity in technical effort: during a major technological shift, the marginal cost of R&D may rise with the scale of the program (architectural complexity, refactoring, technical debt), or fall if learning effects dominate. We then add three interaction terms that map into structural and fairly persistent coordination channels: $RnD\times$ *salesforce* (product–market alignment and roadmap coordination with sales), $RnD \times AT$ (coordination between development and support, especially as the installed base and incidents grow), and $Marketing\times$ *salesforce* (the acquisition-to-conversion chain). By contrast, we do not include the full set of possible cross-terms between departments: in short samples, they tend to be highly collinear, inflate coefficient variance, and generate effects that are difficult to attribute to stable organizational mechanisms. Appendix E reports, as a complement, the coefficients from the full translog for readers who wish to inspect the entire second-order structure beyond this guided interpretation.

Estimation is carried out via Ridge, in line with the guided nature of the specification and the goal of retaining all included terms, with λ selected by blockwise time-series cross-validation and the *1-SE* rule. Coefficient uncertainty is assessed through a moving-block bootstrap (four-quarter blocks) to preserve short-run serial dependence and deliver empirical standard errors (SE_{boot}). Sign-robustness diagnostics (Appendix F) complete the interpretation: the majority sign is the most frequent sign of the coefficient across bootstrap replications, and the majority share is the fraction of replications displaying that sign. These indicators capture directional stability (positive vs. negative) without conflating it with standard asymptotic significance.

Summary reading of the results

Table 6: Guided Translog (Ridge): coefficients and SEboot

	Desk		Desk + SaaS		Desk + SaaS + Full Web	
Term	Coef	SE_boot	Coef	SE_boot	Coef	SE_boot
revenue	0.015	0.010	0.013	0.034	-0.007	0.015
units	-0.086	0.006	-0.084	0.015	-0.002	0.029

RnD	0.094	0.004	0.100	0.019	0.128	0.014
marketing	0.051	0.024	0.063	0.011	0.055	0.007
salesforce	0.114	0.004	0.082	0.014	0.058	0.027
Others	0.124	0.008	0.038	0.014	0.100	0.023
tech_support	0.095	0.007	0.097	0.023	0.035	0.026
I(RnD:RnD)	0.014	0.001	0.015	0.003	0.018	0.002
I(marketing:salesforce)	0.018	0.003	0.019	0.003	0.015	0.002
I(RnD:salesforce)	0.015	0.001	0.023	0.004	0.044	0.006
I(RnD:tech_support)	0.016	0.001	0.017	0.002	0.037	0.005

Two results stand out. First, the curvature in R&D—captured by the quadratic term $I(RnD{:}RnD)$—is positive and strikingly stable across the three phases ($0.014 \to 0.015 \to 0.018$), with small bootstrap standard errors. This convexity implies that scaling up R&D effort is associated with a rising marginal cost. The pattern is consistent with increasing complexity (rewrites, industrialization, higher service-quality requirements, security, and compliance) and suggests that, over the period studied, learning-by-doing is not strong enough to offset the accumulation of technical and organizational constraints.

Second, the interaction terms are systematically positive and highly robust, but their magnitude varies with the technological regime. The $RnD : salesforce$ coupling rises sharply across phases ($0.015 \to 0.023 \to 0.044$), indicating that as the firm moves toward *Full Web*, product–market coordination becomes more expensive—through tighter roadmap alignment, dedicated sales resources, and prioritization trade-offs under customer constraints. Likewise, $RnD{:}AT$ increases markedly in the last phase ($0.016 \to 0.017 \to 0.037$), suggesting that the technological shift strengthens the link between development and support: higher operational requirements, incidents, compatibility constraints, migration assistance, and version management in production. By contrast, the $Marketing : salesforce$ interaction remains positive and broadly stable ($0.018 \to 0.019 \to 0.015$), pointing to a relatively continuous acquisition-to-conversion chain, without a reorganization comparable to what is observed around R&D.

On the linear terms, the guided translog does not overturn the earlier economic hierarchy; it sharpens it. R&D remains a robust cost driver (positive coefficients and a 100% positive sign share in every phase), while the elasticity associated with volume (*units*) is negative on average but becomes less precisely pinned down under *Full Web* (larger SE_{boot} and a majority negative sign share of 86.8%). This is consistent with the idea that "dilution through volume" fades in the last regime. Revenue (*revenue*) shows weaker sign stability throughout, reinforcing the interpretation of this variable as a value/price proxy (product mix, pricing policy, moving upmarket) rather than a direct measure of physical output.

Overall, the translog refines the diagnosis on R&D: the rising contribution of R&D to average unit cost is not only a level effect (more spending) but also a structural one. On the one hand, the positive quadratic term implies convexity: the marginal cost of intensifying R&D increases, consistent with diminishing marginal returns to technical effort as complexity and industrialization constraints build up. On the other hand, the strengthening of the RnD:*salesforce* and $RnD{:}AT$ interactions—roughly three times larger in the last phase than in the first—shows that R&D is increasingly accompanied by downstream cross-costs. Value creation becomes more dependent on product–market articulation (selling a service, migrations, roadmap alignment) and product–operations articulation (support, incidents, service quality). Put differently, the growing weight of R&D in unit cost reflects both (i) a rising marginal technical burden and (ii) an intensification of the organizational complementarities and frictions that R&D induces—two mechanisms that naturally align with the *SaaS* and then *Full Web* transition.

Table 7: Guided Translog (Ridge): fit quality by phase (in-sample and blocked CV, 1-SE rule)

Term	Desk	Desk + SaaS	Desk + SaaS + Full Web
%Var (in-sample)	99.845	99.520	98.520
RMSE (in-sample)	0.010	0.010	0.013
%Var (blocked CV, 1-SE)	99.704	97.197	97.579
RMSE (blocked CV, 1-SE)	0.014	0.026	0.016
lambda (1-SE)	0.048	0.033	0.026
N	18.000	18.000	20.000

Model fit is strong in every phase, including under block cross-validation using the 1-SE rule. Cross-validated metrics remain very favorable in the *Desk* phase and deteriorate moderately in *Desk + SaaS* (higher CV RMSE), which is expected in a transition period where multiple operating logics coexist. The selected λ declines over time ($0.048 \rightarrow 0.033 \rightarrow 0.026$), indicating that in the most recent subsample the amount of shrinkage required to stabilize coefficients is slightly lower for this guided specification—without weakening the case for regularization, since the core diagnostic here concerns signs and the economic coherence of the interaction structure.

6.2 Comparing specifications and interpreting the evidence

The regularized Cobb–Douglas and the guided translog converge on the central message about returns to scale, but they do so at different levels of granularity. Cobb–Douglas delivers an immediately operational summary: the sum of elasticities, Σ_β, lies below one in *Desk* (0.712), falls further in *Desk + SaaS* (0.626), and then rises above one in *Desk + SaaS + Full Web* (1.041). That is sufficient to establish a shift from increasing returns to scale to decreasing returns to scale—meaning that unit cost no longer mechanically declines with activity—once *Full Web* becomes a defining regime. The guided translog does not challenge this conclusion; it qualifies it by suggesting that the regime change is less about curvature in outputs per se and more about internal nonlinearities and cross-costs between functions that become salient as architecture and operations grow more complex.

Differences in coefficients across the two specifications reflect a change in decomposition rather than a contradiction. Cobb–Douglas provides a clean and readable functional hierarchy (for instance, the R&D contribution rises sharply in the *Full Web* phase), but each coefficient blends (i) a level effect and (ii) coordination, complementarity, or saturation effects. The guided translog disentangles these components: the R&D quadratic term is positive and stable ($0.014 \rightarrow 0.015 \rightarrow 0.018$), indicating rising marginal costs as R&D effort intensifies; and, crucially, R&D-related interactions (*RnD*:*salesforce* and *RnD*:*AT*) strengthen substantially. Viewed through the guided translog, the increase in the “weight” of R&D seen in Cobb–Douglas can therefore be read as the joint outcome of a direct effect (more R&D spending) and an increase in the cross-costs it generates with sales and support—especially in the *Full Web* regime, where a larger fraction of downstream spending is tied to coordination with technical effort.

In terms of contribution and use, the regularized Cobb–Douglas is best suited to the headline message and to a managerial reading: it stabilizes elasticities, facilitates comparisons across phases, and provides the synthetic Σ_β indicator for assessing returns to scale. The guided translog plays a complementary, more analytical role: it is not necessary to establish the regime shift, but it is valuable for documenting micro-organizational mechanisms consistent with that shift (technical saturation and rising inter-functional frictions). This sequencing—robust diagnosis via Cobb–Douglas, mechanism-oriented interpretation via guided translog—is what clarifies the paper's contribution to the literature.

7 Discussion and conclusion

Tracking a French software publisher over fourteen years—from an initial *Desk* model to the introduction of *SaaS*, and then the rollout of a *Full Web* architecture alongside the legacy stack—shows that "strong return to scale" are far from automatic. Ridge estimates first reveal a regime of strong increasing returns to scale: as long as SaaS scales without a major pivot into *Full Web* rewriting, the elasticity sum Σ_β remains well below one (roughly 0.25 to 0.70), indicating that unit costs are primarily diluted as activity expands. This pattern reverses once *Full Web* investment crosses a threshold: Σ_β rises durably above one and the firm temporarily enters a decreasing-returns regime in which costs grow faster than volume.

The guided translog extension estimated by Ridge corroborates this reversal and sheds light on its mechanisms. Positive curvature in R&D points to rising marginal technical costs as the transformation intensifies—consistent with greater complexity (industrialization, reliability requirements, technical debt, coordination). Positive interactions between R&D and downstream functions (notably *AT* and *salesforce*) are also consistent with the view that technological change raises coordination costs: as the publisher enters the *Full Web* phase, commercial performance and support quality become more tightly linked to technical trade-offs, which can erode the "mechanical" dilution of fixed costs one might otherwise expect. In this sense, the translog is not meant to replace Cobb–Douglas as the reference model; rather, it provides a complementary lens by making visible nonlinearities and cross-costs that can rationalize the regime change documented by Σ_β.

The managerial implication is straightforward: even sizable return to scale can be offset by a wave of transformation investment when R&D—and, more broadly, the associated operating and coordination effort—grows faster than the firm's capacity to dilute costs through volume. The issue is not only "growing," but synchronizing the investment path with adoption and monetization, so that technical effort remains proportionate to the activity served.

Several governance implications follow for software publishers undergoing technological transitions. First, Σ_β can serve as a compact indicator for managing investment phases: a persistent move above one signals that transformation temporarily dominates scale gains. Second, alignment between development effort and commercial dynamics is critical: scale gains materialize only if the pace of rewriting, industrialization, and feature expansion remains consistent with the effective growth of the user base. Third, budget discipline can build on this diagnosis: reinvesting scale gains during favorable phases can smooth investment peaks and limit pressure on marginal profitability during regime shifts. Finally, the presence of cross-costs (translog interactions) argues for complementing department-level KPIs with inter-functional coordination indicators, to detect organizational frictions early—before they translate into a sustained rise in unit costs.

This study is constrained by the short length of each segment (18 to 20 quarters per phase) and by data confidentiality, which prevents full public release of the underlying dataset. Bootstrap corrections and robust inference mitigate—but cannot fully eliminate—small-sample concerns. External validity would benefit from replication on a multi-firm panel, with a richer treatment of pricing and competition dynamics and a more granular product-level decomposition to separate range effects, mix effects, and architecture effects.

Even so, in the case studied, the software industry does exhibit strong return to scale—reinforced by the rise of *SaaS*—yet a major technological transformation such as a *Full Web* rewrite can temporarily neutralize them. The same lens naturally extends to current investment cycles in artificial intelligence: integrating generative models, building MLOps pipelines, scaling compute and storage, and meeting higher security and compliance requirements represent an investment wave that can reconfigure cost structures and, at least transiently, reshape returns to scale. As with the shift to *Full Web*, these investments are not merely technical outlays: they deepen interdependencies across development, operations, support, and sales, and can raise coordination costs before productivity gains, usage standardization, and large-scale diffusion catch up. More broadly, neither *Full Web* nor AI eliminates fixed costs; they shift the frontier and create new ones. Architecture quality, control of technical debt, and cross-functional coordination—rather than volume growth alone—emerge as key conditions for achieving durably increasing returns to scale.

8 References

1. Eurostat. (2025). *Businesses in the information and communication services sector.* Statistics Explained (Eurostat).
2. Gartner. (2024). *Gartner Forecasts Worldwide IT Spending to Grow 6.8% in 2024.* Press release (Gartner).
3. Gartner. (2025). *Gartner Forecasts Worldwide IT Spending to Grow 7.9% in 2025.* Press release (Gartner).
4. Varian, H. R. (2023). *Economics of digital goods revisited. Journal of Economic Perspectives, 37*(2), 145–170.
5. Boehm, B. W. (1981). *Software engineering economics.* Englewood Cliffs, NJ: Prentice Hall.
6. Putnam, L. H. (1978). A general empirical solution to the macro software sizing and estimating problem. *IEEE Transactions on Software Engineering, 4*(4), 345–361.
7. Abdel-Hamid, T. K., & Madnick, S. E. (1991). *Software project dynamics: An integrated approach.* Englewood Cliffs, NJ: Prentice Hall.
8. Boehm, B. W., Abts, C., & Brown, A. W. (2000). *Software cost estimation with COCOMO II.* Upper Saddle River, NJ: Prentice Hall.
9. Banker, R. D., & Slaughter, S. A. (1997). A field study of scale economies in software maintenance. *Management Science, 43*(12), 1709–1725.
10. Lientz, B. P., & Swanson, E. B. (1980). *Software maintenance management.* Reading, MA: Addison-Wesley.
11. Wiederhold, G. (2006). *What is your software worth? Communications of the ACM, 49*(9), 85–87.
12. Kaplan, R. S., & Anderson, S. R. (2004). Time-driven activity-based costing. *Harvard Business Review, 82*(11), 131–138.
13. Ooi, G., & Soh, C. (2003). *Developing an activity-based costing approach for system development and implementation. ACM SIGMIS Database, 34*(3), 54–71. ([Core][3])
14. Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics, 12*(1), 55–67.
15. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 58*(1), 267–288.
16. Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67*(2), 301–320.
17. Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software, 33*(1), 1–22.
18. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). New York, NY: Springer.
19. Künsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *Annals of Statistics, 17*(3), 1217–1241.
20. Lahiri, S. N. (2003). *Resampling methods for dependent data.* New York, NY: Springer.
21. Chow, G. C. (1960). Tests of equality between sets of coefficients in two linear regressions. *Econometrica, 28*(3), 591–605.
22. Andrews, D. W. K. (1991). Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica, 59*(3), 817–858.
23. Newey, W. K., & West, K. D. (1994). Automatic lag selection in covariance matrix estimation. *The Review of Economic Studies, 61*(4), 631–653.
24. Breusch, T. S., & Pagan, A. R. (1979). A simple test for heteroscedasticity and random coefficient variation. *Econometrica, 47*(5), 1287–1294.

25. Ljung, G. M., & Box, G. E. P. (1978). On a measure of lack of fit in time series models. *Biometrika, 65*(2), 297–303.
26. Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association, 74*(366), 427–431.
27. Phillips, P. C. B., & Perron, P. (1988). Testing for a unit root in time series regression. *Biometrika, 75*(2), 335–346.
28. Kwiatkowski, D., Phillips, P. C. B., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics, 54*(1–3), 159–178.
29. Cobb, C., & Douglas, P. H. (1928). A theory of production. *The American Economic Review, 18*(1), 139–165.
30. Christensen, L. R., Jorgenson, D. W., & Lau, L. J. (1973). Transcendental logarithmic production frontiers. *The Review of Economics and Statistics, 55*(1), 28–45.
31. Shephard, R. W. (1953). *Cost and production functions.* Princeton, NJ: Princeton University Press.
32. Jackson, R. (1988). A museum cost function. *Journal of Cultural Economics, 12*(1), 41–50.
33. Gillen, D. W., Oum, T. H., & Tretheway, M. W. (1990). Airline cost structure and policy implications: A multi-product approach for Canadian airlines. *Journal of Transport Economics and Policy, 24*(1), 9–34.
34. Jara-Díaz, S., Martínez-Budría, E., Cortés, F., & Basso, L. (2002). A multioutput cost function for the services of Spanish ports' infrastructure. *Transportation, 29*, 419–437.
35. Bagadeem, S. (2021). Estimating the translog cost function for the telecommunications sector in Saudi Arabia: The case of Saudi Telecom Company. *International Journal of Financial Research, 12*(3).
36. Autorité de la concurrence. (2023). *Avis n° 23-A-08 du 29 juin 2023 relatif au fonctionnement de la concurrence dans le secteur de l'informatique en nuage (cloud).*
37. Manganelli, A., & Schnurr, D. (2024). *Competition and regulation of cloud computing services: Economic analysis and review of EU policies.* Centre on Regulation in Europe (CERRE), Research Report.
38. Brynjolfsson, E., & Hitt, L. M. (1997). Information technology and internal firm organization: An exploratory analysis. *Journal of Management Information Systems, 14*(2), 81–101.
39. Griliches, Z. (1979). Issues in assessing the contribution of research and development to productivity growth. *The Bell Journal of Economics, 10*(1), 92–116.
40. Griliches, Z. (1986). Productivity, R&D, and basic research at the firm level in the 1970s. *American Economic Review, 76*(1), 141–154.
41. Wald, A. (1943). Tests of statistical hypotheses concerning several parameters when the number of observations is large. *Transactions of the American Mathematical Society, 54*(3), 426–482.
42. Gujarati, D. N., & Porter, D. C. (2009). *Basic econometrics* (5th ed.). New York, NY: McGraw-Hill.

9 Appendix

9.1 A. Allocation of Operating Costs by Department

1. Scope Operating costs include:

- General overheads (e.g. subscriptions, energy, insurance, maintenance, transportation, etc.)
- Miscellaneous expenses (e.g. sundry purchases, commissions, packaging, documentation, etc.)
- Travel & mobility (tickets, fuel cards, tolls, etc.)
- Marketing expenses (advertising, public relations, promotional items, point-of-sale materials, etc.)
- Outsourcing (external services, fees, training, etc.)
- Internal software (development and maintenance costs)
- SaaS/Web costs & royalties

2. Allocation methodology

- Marketing expenses: fully allocated to the Marketing Department.
- Outsourcing: directly allocated to the department using the service.
- General overheads, miscellaneous expenses, travel and mobility: allocated in proportion to the number of employees in each department, according to the following formula:

$$\text{Cost allocated to } \mathrm{D}_i = \frac{\text{Number of employees in } \mathrm{D}_i}{\text{Total number of employees}} \times \text{Total cost}$$

- SaaS/Web costs & royalties: treated as model noise; not allocated within the scope of this perimeter.

9.2 B. Stationarity Tests for Log-Differenced Variables

For each variable, three complementary tests are applied:

- **ADF** (Augmented Dickey–Fuller) — H_0: unit root.
- **PP** (Phillips–Perron, Z-α statistic) — stationarity if the ratio |stat| / |5% critical value| < 1.15.
- **KPSS** — H_0: stationarity around a constant.

The series cover the full study period and are transformed into logarithms and then first-differenced in order to preserve their economic interpretation (growth rates). A constant is included in the test regressions, and the number of lags is selected according to the Schwarz criterion.

series	variable	ADF_stat	ADF_p	KPSS_stat	KPSS_p	PP_ratio
level	unit_cost	-2.520	0.365	0.135	0.100	0.791

level	revenue	-2.167	0.507	1.469	0.010	0.071
level	units	-3.190	0.098	1.469	0.010	0.125
level	RnD	-1.823	0.646	0.793	0.010	0.647
level	marketing	-2.629	0.321	0.789	0.010	0.513
level	salesforce	-2.917	0.205	1.383	0.010	0.113
level	Others	-1.528	0.765	0.207	0.100	0.384
level	tech_support	-0.617	0.972	0.425	0.067	0.199

For the eight series in levels, the tests are broadly consistent:

- **ADF** never rejects the unit-root null hypothesis (p-values above 5%).
- **KPSS** rejects stationarity for all series, except possibly *unit_cost* and *Others*, where the statistic remains just at the 10% threshold.
- **PP** yields ratios below 1.15, but the ADF and KPSS tests, which are more stringent in this context, point toward non-stationarity.

9.3 C. Timeline

Table 9: Timeline of revenue shares and Research budget allocation (percentages)

	Share of total revenue			Share within the R&D budget	
Period	OnPremise	SaaS	Fullweb	Other	Fullweb
2011	100	–	–	100	–
-	100	–	–	100	–
2014	98	2	–	100	–
-	-	-	–	100	–
2017	92	8	–	100	–
2018	89	11	–	100	–
2019	84	16	–	100	–
2020	78	22	–	97	3
2021	70	30	–	91	9
2022	64	35.98	0.02	85	15
2023	58	41.66	0.34	79	21
2024	57	42	1	73	27

9.4 D. Bootstrap by phase

Table 10: Appendix D — Moving-block bootstrap (Ridge): Coef, SE-boot and pboot (blocks=4, B=1000)

	Desk			Desk + SaaS			Desk + SaaS + Full Web		
Term	Coef	SE_boot	p_boot	Coef	SE_boot	p_boot	Coef	SE_boot	p_boot
(Intercept)	3.5547	0.3735	0.00	4.0255	0.7425	0.000	1.6925	1.0937	0.040
revenue	0.0024	0.0098	0.91	-0.0010	0.0501	0.410	-0.0197	0.0259	0.790

units	-0.1362	0.0204	0.00	-0.1543	0.0258	0.000	0.0027	0.0589	0.706
RnD	0.2047	0.0124	0.00	0.2666	0.0347	0.000	0.4639	0.0236	0.000
marketing	0.1108	0.0297	0.00	0.1333	0.0257	0.000	0.1174	0.0178	0.000
salesforce	0.2322	0.0202	0.00	0.1388	0.0239	0.000	0.1791	0.0551	0.000
Others	0.1410	0.0214	0.00	0.0480	0.0177	0.064	0.1460	0.0266	0.004
tech_support	0.1570	0.0145	0.00	0.1949	0.0319	0.000	0.1519	0.0543	0.016

9.5 E. Full Translog

Table 11: Appendix E — Full Translog (Ridge): coefficients by phase

Term	Desk	Desk + SaaS	Desk + SaaS + Full Web
(Intercept)	3.6520	4.2068	3.5974
revenue	0.0025	-0.0013	-0.0041
units	-0.0220	-0.0127	-0.0012
RnD	0.0219	0.0177	0.0257
marketing	0.0220	0.0131	0.0100
salesforce	0.0229	0.0118	0.0123
Others	0.0211	0.0014	0.0158
tech_support	0.0212	0.0115	0.0111
0.5*revenue^2	0.0002	-0.0001	-0.0002
0.5*units^2	-0.0020	-0.0011	-0.0001
0.5*RnD^2	0.0064	0.0053	0.0071
0.5*marketing^2	0.0084	0.0050	0.0042
0.5*salesforce^2	0.0064	0.0036	0.0046
0.5*Others^2	0.0072	0.0005	0.0054
0.5*tech_support^2	0.0076	0.0036	0.0038
marketing×Others	0.0055	0.0020	0.0033
marketing×salesforce	0.0046	0.0034	0.0028
marketing×tech_support	0.0051	0.0029	0.0027
Others×tech_support	0.0039	0.0039	0.0029
revenue×marketing	0.0014	0.0008	0.0006
revenue×Others	0.0012	0.0001	0.0011
revenue×RnD	0.0014	0.0010	0.0012
revenue×salesforce	0.0014	0.0009	0.0011
revenue×tech_support	0.0013	0.0006	0.0009
revenue×units	-0.0011	-0.0003	-0.0001
RnD×marketing	0.0044	0.0031	0.0040
RnD×Others	0.0038	0.0029	0.0064
RnD×salesforce	0.0032	0.0038	0.0090
RnD×tech_support	0.0036	0.0023	0.0079
salesforce×Others	0.0036	0.0010	0.0031
salesforce×tech_support	0.0036	0.0034	0.0022
units×marketing	0.0016	0.0010	0.0009
units×Others	0.0022	0.0000	0.0018
units×RnD	0.0029	0.0014	0.0018
units×salesforce	0.0031	0.0015	0.0018
units×tech_support	0.0024	0.0009	0.0017

9.6 F. Sign robustness (Guided Translog)

Table 12: Appendix F — Sign robustness (moving-block bootstrap, majority sign and share)

	Desk	Desk + SaaS	Desk + SaaS + Full Web

Term	Major_sign	Major_share(%)	Major_sign	Major_share(%)	Major_sign	Major_share(%)
revenue	+	88.6	-	71.2	-	70.8
units	-	100.0	-	100.0	-	84.2
RnD	+	100.0	+	100.0	+	100.0
marketing	+	97.6	+	100.0	+	100.0
salesforce	+	100.0	+	100.0	+	100.0
Others	+	100.0	+	98.6	+	100.0
tech_support	+	100.0	+	100.0	+	98.2
‘I(RnD:RnD)‘	+	100.0	+	100.0	+	100.0
‘I(marketing:salesforce)‘	+	100.0	+	100.0	+	100.0
‘I(RnD:salesforce)‘	+	100.0	+	100.0	+	100.0
‘I(RnD:tech_support)‘	+	100.0	+	100.0	+	100.0