Transaction Costs and Outsourcing Dynamics: A System Dynamics Approach

Edoardo Mollona
Università degli Studi di Bologna
Mura Anteo Zamboni, 7, 40127 Bologna, Italy
emollona@cs.unibo.it

Alessandro Sposito
Università degli Studi di Bologna
Via Murri 29, 40137 Bologna, Italy
sposito@cs.unibo.it

ABSTRACT

Increasingly, firms are outsourcing their processes. Outsourcing regards not only peripheral activities but portions of key activities as well. Outsourcing is a pervasive phenomenon that often overlaps, and interferes with, the strategic decision to define a firm's boundaries. By outsourcing phases of R&D processes, for example, firms may lose their know-how concerning the production of core products. The outsourcing of processes may interact with a firm's organizational learning, thereby influencing the definition of the boundary that includes the core activity of the firm. Thus, managing outsourcing requires regulating the flow of knowledge that leaves the organization. Such a calibration produces transaction costs, which, in the long term, may decrease the desirability of outsourcing. To address the dynamic interplay among forces at work in outsourcing processes we built a system dynamics model. Specifically, the model represents the different components of software production. One of the main industries on which outsourcing has had a big impact in the last few years is software development. Our model attempts to include the main dynamics of a software-house company that is considering the possibility of outsourcing its production. We used the model to generate a number of hypotheses to study long term consequences of outsourcing policies.

INTRODUCTION

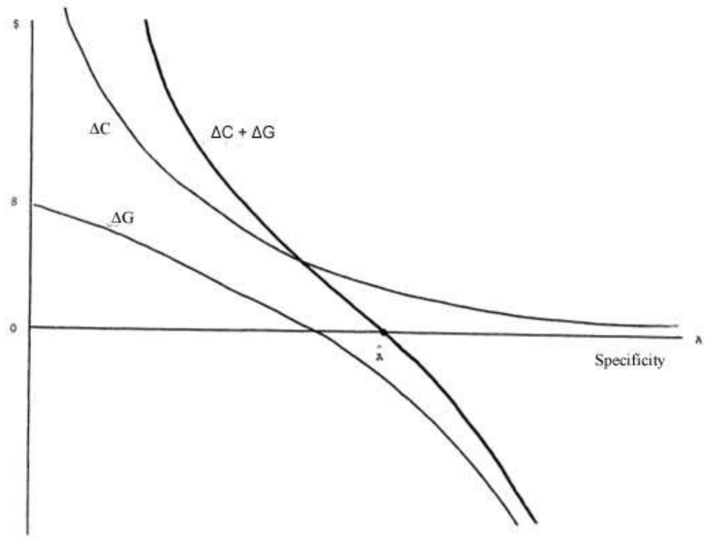
In order to create this model, we based our assumptions and concepts on Transaction Cost Economics. This theory was developed by R. Coase [COA37] and O. Williamson [WIL81]. Their studies concern the nature of the firm, the idea of transaction, and the importance of understanding how to manage these activities to better understand the form of a company.

Each transaction with the market has a cost. Companies exist so that their structures can better and more efficiently coordinate those transactions with the market. Different researchers have studied this topic. Commons defined a transaction as that elementary unit through which economic activity is made, which means that the exchange between actors of the market is considered the fundamental and basic economic activity [COM34]. Later, Williamson defined a transaction as the transfer of a good or a service through a technologically separated interface, which includes a value exchange between the actors [WIL75].

This means that if the market were responsible for managing all the transactions, they would be more costly, especially because of lack of information. Companies are created and expanded with the intent of minimizing transaction costs, by internalizing some exchanges previously regulated by the market. The goal of each company is to reduce total costs, defined as the sum of production costs and governance costs.

The production costs are those costs strictly related to creating the company's output. The governance costs, instead, are all of the expenses that need to be covered in order to manage the transactions and to manage the company. Transactions costs are tied to information costs, market costs, selling costs, and legal costs. Managing costs are those necessary to maintain relationships with providers and customers and to manage the employees.

These concepts have been applied when creating the model and the strategies defined by the simulations' results are primarily based on the ratio between total costs of a traditional production and total costs of outsourced production.



Governance costs and production costs when specificity increases

O.E. Williamson, The Economics of Organization: The Transaction Cost Approach, *The American Journal of Sociology*

The Williamson theory [WIL75] also talks about three main variables that set the transactions' behavior:

- Frequency
- Uncertainty
- Asset specificity

We considered all of these in the structure of the model, with special reference to the last one. Software specificity has been used to determine the availability of products on the market, to regulate the power of providers, and to better explain the concept of software modularity.

In particular, the history of the software industry has cases in which the connection between software specificity and modularity has been well defined. IBM 360, for example, was a platform that was able to satisfy multiple customers' needs, thanks to the possibility of including and combining different software modules [LAN02]. This meant low specificity but it also meant the possibility for other competitors to produce software for the 360 platform, possibly reducing the IBM leadership.

More generally, the possibility of dividing a project into multiple parts and developing these subparts at the same time, partially inside the company and partially through suppliers, depends on the project modularity [ULL00]. If the software project is planned and designed to be divided into multiple modules that can be assembled together, the firm can more easily and more efficiently be oriented towards outsourcing [SUGRHA].

THE MODEL

The model is best described by starting with one of the three stock variables, "Fraction of Outsourcing." This variable represents the most relevant index used to define the best strategy that a company should adopt. The "Fraction of Outsourcing" variable indicates the outsourced portion of the company's total production. The initial value is defined by the parameter "Unit Fraction of Outsourcing," which allows the running of a simulation with reference to companies that already outsource their production in different percentages. The "Fraction of Outsourcing" is also helpful in identifying the optimal value of off-shore production at which to start, in order to reduce both the long and short term costs.

The parameter "Time to Outsource" is the amount of time that the company needs to increase or reduce the percentage of production outsourced to third parties, when implementing a change of strategy. The parameter is expressed in months and basically sets the delay between the moment in which the new strategy is decided and the moment in which the new strategy is fully adopted.

Working our way backwards in the model, the parameter "Time to Decide" represents the delay related to the company's decision-making process. In addition to the time required to apply a new strategy, it is important to consider the time invested in making the decision to adopt a new strategy. Once a company finds out that there is a new optimal equilibrium between in-house and outsourced production, the choice of a new production plan could be influenced and delayed due to the organizational structure's characteristics, the finance availability, and the managers' decision-making power.

"Desired Outsourcing" is then a stock variable that clearly represents the portion of outsourcing that the company wants to use in its production; it is considered the best choice at that moment. System Dynamics delay characteristics are appropriate in this case, as they leave room for a progressive set of actions that the company would undertake both when deciding to adopt and to apply a new strategy.

The presence of a double delay, rather than single, better reproduces an organizational structure and its decision-making process, making the model more accurate. In addition to this, having the possibility of setting two different delays allows us to use the model to simulate various kinds of companies and multiple market segments that can be characterized by different production rhythms.

Moving on from "Fraction of Outsourcing," the first feedback loop begins with the variable "Software Modularity." This factor represents the chance of dividing a software project into multiple parts. We define a component as a physically distinct portion of a product, which embodies a core design concept and performs a well-defined function [HENCLA90]. A company that wants to outsource part of its production needs to be able to merge modules developed with different technologies by multiple providers. "Software Modularity" expresses the firm's ability to manage outsourced projects. This means having a better knowledge of how to coordinate products developed by a multiform network of providers and merge them, through an integrating process, into one product which is the total project [BRUPREPAV01].

In this model, the more the company outsources, the better its ability to optimize this production phase is. The idea is that if the company has the tendency to outsource, then most likely the projects will be thought as dividable and modular.

The capacity of efficiently managing partially outsourced projects has a direct impact on the production costs. Since this study wants to compare the dynamics of traditional production and outsourcing production, the production costs of both strategies have been included in the model.

Some of the costs that need to be covered when producing with outsourcing are called costs of integration. These make up all of the firm's production costs, excluding the labor costs and the material costs. The costs of integration are related to, for example, the information exchange, the management of external contacts, the interpretation and the understanding of what has been produced in outsourcing, and the assembling of modules written by the multiple providers.

Costs of integration are an important aspect when evaluating whether producing with outsourcing is more or less convenient than an in-house production [ANDPAR00] and they represent the variable part of total production costs. Labor costs, which in the software industry can be quantified in hourly fares or cost per line of code, could be considered as a constant for a specific technology, since they are determined by the market and are mostly the same for each company. Costs of integration refer to the transformation process, which, in the last few years, has been changing the firms into integrators rather than producers [ANDPAR02].

In the feedback loop, the easier it is to divide the software project into multiple modules, the less the production of some of these modules in outsourcing will cost.

"Software modularity" is also connected, in another feedback loop, to the variable "Sw Specificity," a concept inherited from the Williamson theory [WIL81]. This variable increases or decreases the purchasing cost for software produced in outsourcing depending on the availability of that type of software code in the market place. The more specific and unique the software is, the more costly it will be to purchase it from external providers. This is a direct consequence of the economy of scale theory. Standard code is often developed by more companies and, therefore, easier to find on the market. In the same way, a software module with high specificity requires a custom development and, presumably, higher programming skills.

When a project is designed to be divided into different parts, some of which will be developed through outsourcing, the idea is that these outsourced modules can be found on the market easily and at a cheap price. That is why there is a negative correlation between “Software modularity” and “Sw specificity.”

As explained, the specificity and the level of uniqueness of the project code determine the purchasing cost of software modules from third parties. This price is represented by the variable “Cost of Outsourced Product.” The sum of this variable and “Cost of integration” gives the total costs of outsourcing, a value indicated by the variable “Outsourcing costs.”

In order to better explain how the production strategies are compared within this model, it is appropriate to move to a different loop before completing the one thus far discussed. The stock variable “Fraction of Outsourcing” is directly connected to the third and last stock, “Cum Units Insourced,” which represents the quantity of code that, at each time step of the simulation, is produced inside the company. Through its entering flow, this stock variable is expressed in number of lines of code produced, and not with a percentage as is the portion of production outsourced. The loss of experience due to organizational forgetting and the comparison of current and initial insourced production value let the model calculate the know-how level of the firm during the simulation.

The variation of the firm’s experience and know-how has a major impact on the internal production costs. Even if there is a standard fare for developing with a given technology, the employees’ production capacity has a strong influence on the real development cost, since it determines the quality of the project and the time required to complete the software. This is why “Production Cost” depends on the firm’s experience in producing software.

In the model, insourcing and outsourcing costs are compared with each other. As specified in the Williamson theory, the insourced development is based on internal governance costs and production costs. This definition lets us compare the outsourcing and the insourcing strategies at the same level, by analyzing the ratio between market and internal governance costs and the ratio between market and internal production costs.

“Cost of Outsourced Product” and “Production Cost” are, respectively, the outsourced and insourced production costs, whereas “Cost of Integration” and “Int. govern. Costs” are their respective governance costs.

The ratios described in the Williamson theory are included in the variable “Indicated outsourcing,” which, through an auxiliary function, represents the optimal balance between in-house and outsourced production. In this model, we decided to adopt the following rule: If the ratio between total outsourcing costs and total insourcing costs is lower than one, then the model identifies the optimal fraction of outsourcing that should be adopted in the firm’s strategy. If this ratio is equal to one, then it is indifferent to insourcing or outsourcing, so the ideal strategy is to outsource fifty percent of the production. If the ratio is higher than one, it means that producing through external providers is more expensive than developing inside the company, so there is no incentive to outsource and the ideal value for “Fraction of Outsourcing” is zero.

As explained above, the optimal level of outsourcing is adopted into the firm’s strategy only after the decision has been fully made, as expressed by “Time to decide” and “Desired Outsourcing.”

The higher the outsourcing costs, the lower the desired outsourcing; the higher the insourcing costs, the higher the desired outsourcing. Analyzing the signs of each link within the feedback loops thus far described, we see that they are all reinforcing loops (R1, R3 and R5).

Moving on to other parts of the model, the variable “Firm’s learning curve for integration” is also related to the firm’s know-how and ability of producing software. If the company decreases its internal production, it will gain more experience in assembling code written by third parties. On the other hand, increasing the percentage of software developed within the company makes it more difficult to be efficient at integrating modules developed by external providers. There is a negative correlation, then, between the stock “Cum Units Insourced” and the learning curve for integration, as expressed by the negative exponent in the formula [ANDPAR02b]. This effect could be different depending on the market analyzed. This model, through its parameter customization, can be used to reproduce and simulate different kinds of firms and different situations.

The same idea can be applied to the previously described “Production Cost,” which has its own parameter to adapt the model to diverse technologies and market segments. Having different parameters for the learning level of making and putting together gives us the possibility of applying the model to many situations, by combining these factors in different ways. The model is then really flexible at reproducing multivariate companies.

The main effect of a change in the ability of integrating is an increase or a decrease of the costs of integration. If the firm has less experience in putting together software produced by providers, the costs will rise. In the IT world this happens because the outsourced product usually needs to be revised and adjusted, due to changes in the project requirements. The less ability the firm’s employees have in dealing with code written by third parties, the more time and people will be required to make these changes.

In the model, the relationship between the “Firm’s learning curve for integration” and “Cost of integration” is regulated by the parameter “Sensitivity of learning on cost of integration” which, again, gives us the possibility of customizing the model to specific and diverse situations that we might want to analyze. From the “Cost of integration,” the loop is completed with the link already described. The new feedback loop described is also a reinforcing loop (R4).

Being efficient at outsourcing does not only mean having the expertise of integrating the part produced by external partners. Moving toward outsourcing also means completely reviewing the firm’s processes and its way of working. It also requires the capacity of interacting with a new kind of actor in the supply chain and finding out what is the mechanism that makes outsourcing a more convenient producing method, in comparison with in-house development. This requires time and experience, which, in the model, is expressed by the parameter “Learning Outsourcing Properly.” This is the firm’s ability to make the right choices in order to get the most out of the outsourced production. If the fraction of outsourcing increases, the experience in dealing with it also increases. There is then a positive relationship on this link, to show that more experience means more confidence and ability in producing, which, applied to this specific case, means greater capacity to deal with outsourcing [FUNTIR83].

The ability of outsourcing properly is strictly connected with the level of communication with the third parties. The parameter “Communication with supplier” represents the level of quality and intensity with which the firm and its providers exchange information. It has been shown that a better and more precise communication between firms and their providers assures a higher quality for the supply chain [RIP], [MAS93]. In the IT world it is necessary to improve the information exchange to better set the project specifications and requirements and to better select the technologies and the development choices for the software that needs to be produced.

"Learning Outsourcing Properly" and "Communication with supplier" are related by a positive link, since higher experience in outsourcing also means higher quality of communication [SOBSHR98].

As we said, the level of communication has an impact on the supply chain quality. Clearly it also has an impact on the production costs, and in particular on the integration costs. If the experience in outsourcing increases, which means higher levels of communication, the problems related to misunderstanding or bad information exchange will decrease [HEISTU95]. This is why "Communication with supplier" is connected to "Cost of Integration" with a negative link. This relationship closes another feedback loop that has only two negative signs, the one just described and the one between "Outsourcing costs" and "Indicated outsourcing." This feedback loop thus has positive polarity, and it is defined as R2.

It is important to consider that outsourcing production could become a trap in the long term [ANDAND00]. In the technology world, patents and secret company strategies often represent the key to success of a company and are really important in order to achieve a leader position in the market. Exposing the firm's know-how to third parties and revealing details about the projects to the providers could mean, one day, having new competitors. The partner of today could become the rival of tomorrow. Getting knowledge on specific software could lead a provider to enter the same market and try to be independent. Another scenario could see the providers writing code with the same technology and knowledge but for a competitor of the firm. Although this can be legally avoided through detailed partnership contracts, the firm cannot avoid the risk of losing its production secrets through its providers.

On top of this, the main risk is becoming hostage of a set of specific providers. After a long period of collaboration with the same companies, after having invested time and resources to increase the level of communication and having improved the outsourcing process, it is usually not convenient to start the whole process again with different partners. This clearly empowers the providers and amplifies the dependency of the firm on them.

In the model, the higher the level and intensity of communication is, the higher the providers' power is. There is a positive sign on the connection between "Communication with suppliers" and "Suppliers' power" to simulate the need of bigger investments if the firm decides to switch to another provider.

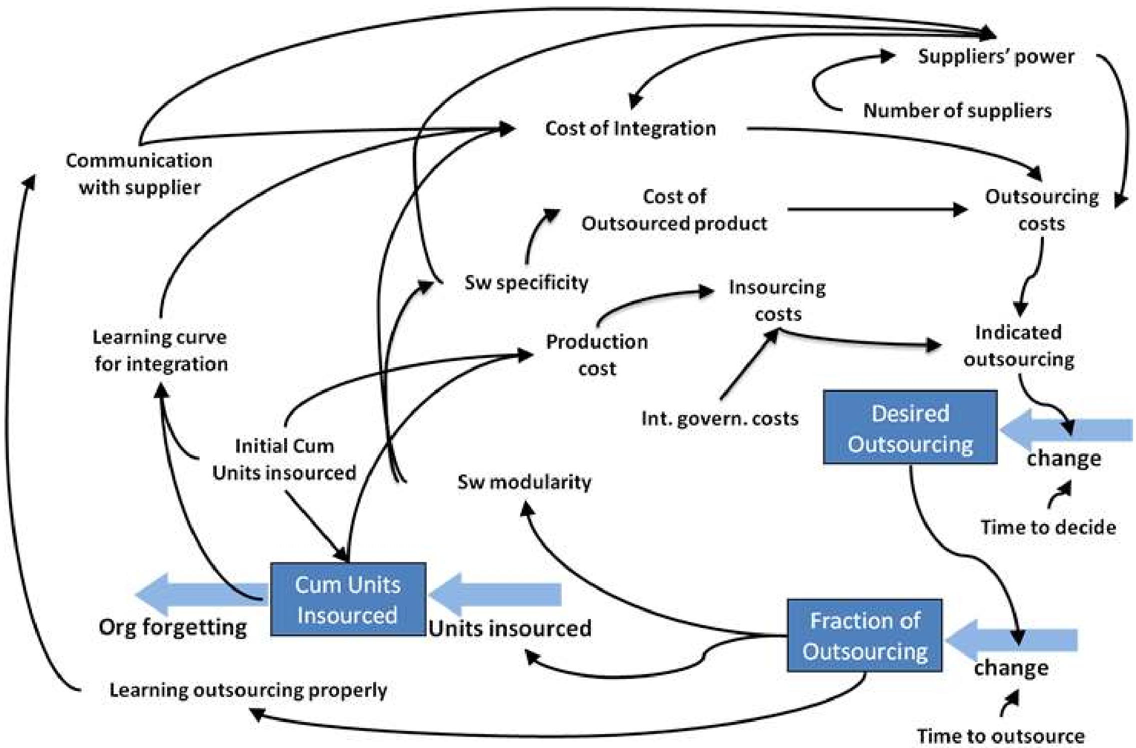
This effect can be magnified or reduced, depending on the number of suppliers with which the firm has partnerships. In fact, a higher number of partners usually means a lower risk of lock-in situations. This explains why the model includes the parameter "Number of suppliers."

If we consider the variable "Sw specificity," we can say that if a firm repeatedly asks the same provider to produce software modules with high specificity, this partner is not able to take advantage of economies of scale and it is tightly dependent on the firm itself. A better and deeper analysis would include a study on the marginal production of each provider for each buyer, but we decided not to focus on it at this stage of the model. However, the literature indicates that the higher the specificity is, the lower the suppliers' power is.

At the same time, though, low specificity means low suppliers' power, since a more standard product is more easily available on the market place, so the firm can choose its partners and switch between them with minimum substitution costs. The function that sets the relationship between "Sw specificity" and "Suppliers' power" in the model assigns the highest power to the providers when the software specificity is not easily found on the market but also not so specific to lock the supplier into that project.

This is economic theory. However, we could say that the complete opposite is also true. Software with high specificity could mean sharing with third parties secrets and patents that the firm owns. For reasons already cited, this could create dependency for the firm, which would tend to keep the same partners and not share its knowledge with new ones. This clearly empowers the suppliers when the software specificity increases. Also, a long relationship could lock the firm to the same partner, because it would be too costly to start a new relationship with different providers.

In this model, though, we decided to follow the economic theory. By changing the function that regulates "Sw specificity" and "Suppliers' power," the model can easily be adapted to different interpretations and this relationship sign inverted.



A sketch of the model

On top of all of these considerations, we could argue that some literature says that higher costs of integration mean higher suppliers' power [ALBSEDSCH03]. A company might, in fact, need more help from the providers if the assembly of software modules produced in outsourcing is not clear and causes problems and delays. The model wants to express this situation through the software specificity as described above. The relationship included thus starts from “Supplier’s power” and ends at “Costs of integration,” to emphasize how higher dependency from the providers means higher governance costs.

With this link we have completed the description of the last two feedback loops present in the model. Both of them have an odd number of negative signs and are, therefore, balancing feedback loops (B1 and B2).

SIMULATIONS

All the simulations that have been run have a clear and constant behavior. Within a certain time period, the model tends to reach an equilibrium which is maintained until the end of the simulation. Acknowledging this particular behavior, we tried to identify which variables were primarily responsible for the equilibrium. We also tried to simulate different strategies and study the simulation results to determine which strategy could be the best for each starting condition that has been reproduced.

The best strategy is the one that can reduce the total costs by decreasing the sum of governance costs and production costs.

Image 1 shows the graph for the stock variable “Fraction of Outsourcing” for two simulations. We consider Firm A and Firm B as two possible companies in the market. Firm A starts with zero percent of outsourcing when the simulation begins. Firm B, instead, has the total production outsourced at time zero. The interesting result is that, within about fifty months, the two companies have the same amount of production outsourced.

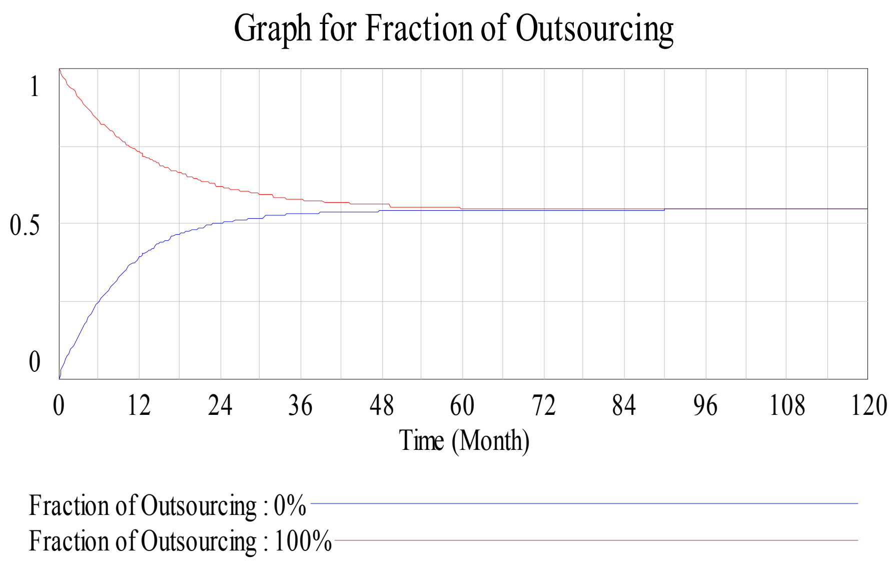


Image 1

Sensitivity simulations have been run to test the model's behavior in different configurations. We modified each parameter independently but we also ran simulations with multiple parameters changed at the same time. Image 2 is the result of a sensitivity simulation in which the learning parameters (for integration, for production, and for outsourcing), the fixed internal production costs, the number of suppliers, and the initial value for "Fraction of Outsourcing" have been changed simultaneously.

Although the model's behavior is always the same, this sensitivity simulation shows that the ideal portion of outsourced production could vary between fifty-two and eighty percent. Other sensitivity simulations showed that modifying the time required to make decisions and to apply a new strategy is very relevant for the output values of the model.

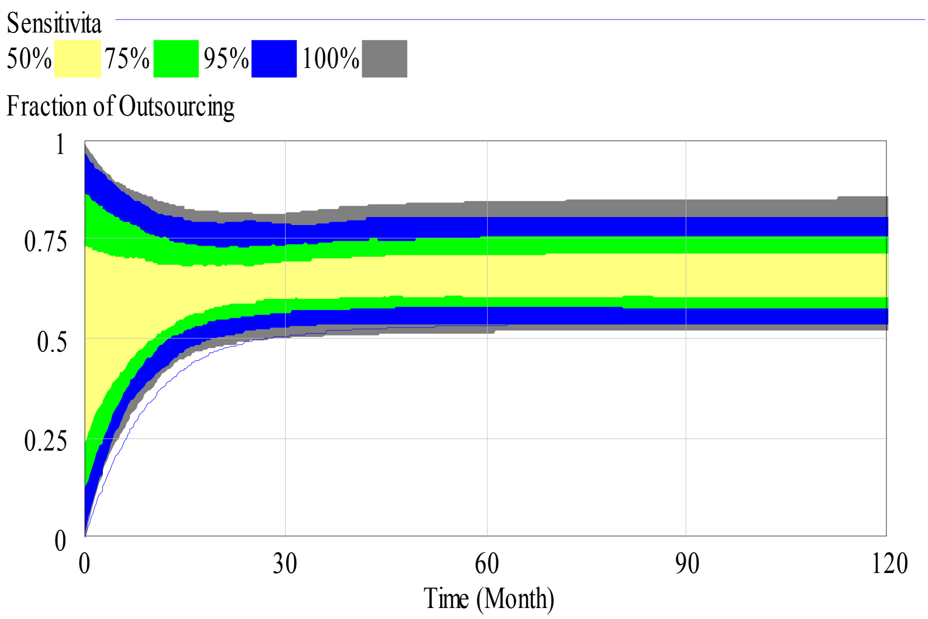


Image 2

In order to perform a cost analysis, the model has been slightly modified and completed. Three stock variables have been added to the described model, so that the insourcing total costs, the outsourcing total costs, and the firm's total cost could all be recorded and monitored during the simulation period.

The first two stocks, at each simulation step, are increased or decreased through their flows, according to the model equations. The third one is the sum of the other two. By analyzing the stock flows, it is possible to see the costs behavior during the simulation period. By analyzing the stocks' values, it is instead possible to determine the total costs and compare different strategies to find the best one, either in the short or in the long term.

With this more complete model structure we tried to find the main factors that reduce the total costs in the short run and in the long run. The model shows that different strategies have significant and measurable effects on total production costs.

The model reacted as expected when modifying certain parameters. A way to reduce the costs of outsourcing, by reducing the costs of integration, is to progressively increase the number of suppliers through which the company outsources. Having more suppliers decreases the suppliers' individual power when deciding the price for the outsourced product. In the model, the suppliers' power has a direct influence on the cost of integration, the governance cost of the outsourced production. In fact, costs of integration can be reduced by using multiple providers in competition with each other.

The model also shows that different industries can behave differently and have different costs. Also, different firms in the same industry could behave differently and have particular ways to approach an outsourcing strategy. We tried to customize these properties in the model through the various learning curve parameters that specify how reactive a firm is in learning how to outsource, learning how to integrate, and learning how to keep producing within the firm itself. By changing these values we simulated that the firm can have a greater or lesser ability of learning how to outsource or that the firm can have a greater or lesser ability of maintaining its production skills. The result is that a high or low tendency to learn can have a strong impact on total production costs (Image 3).

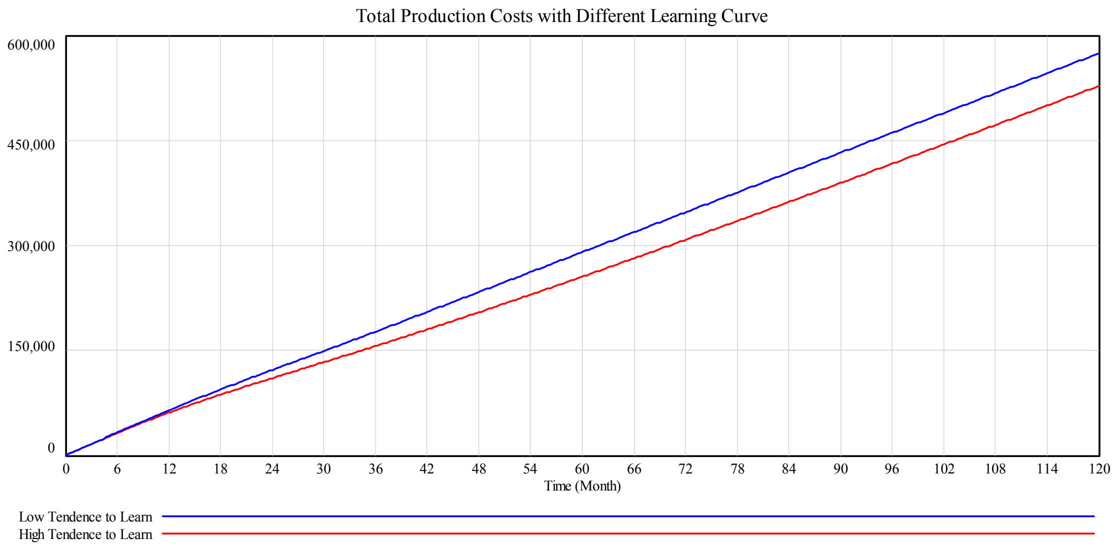


Image 3

Given these results, which helped us validate the correct behavior of the model, we found more interesting results by looking at two main factors of an outsourcing strategy:

- Speed of strategy implementation
- Ability to design modular software.

The first factor refers to the different delays introduced in the model structure. In particular we refer to the delay related to making the decision of outsourcing and to implementing the new strategy of outsourcing. We simulated firms with high ability of switching strategies and moving to or from outsourcing in a reduced amount of time (about 6 months) and firms with a lower ability to do this (about 5 years). Both the delays mentioned above have been increased or decreased to see the model behavior in all conditions.

The second factor refers to the willingness of the firm to move towards outsourcing and to invest energy and organizational changes into it. As described in the previous paragraphs, the more the firm outsources the more the software that is produced should be designed as modular and dividable in parts that can be outsourced. Each firm could be more or less willing to do this and more or less able to learn how to do this. We used two different learning curves to represent how modular the software production could be when the fraction of production given in outsourcing increases or decreases. One of the learning curves represents a more conservative and progressive function, the other one is a more aggressive and proactive way to learn how to design modular software and apply this method to designing software in the organizational structure and procedure.

Having identified these two as main factors that can substantially change the total production costs, we compared the four following strategies that combine different values for the two factors. The different combinations of these two parameters produce the results seen in the four different strategies displayed in the following graphs (Image 4).

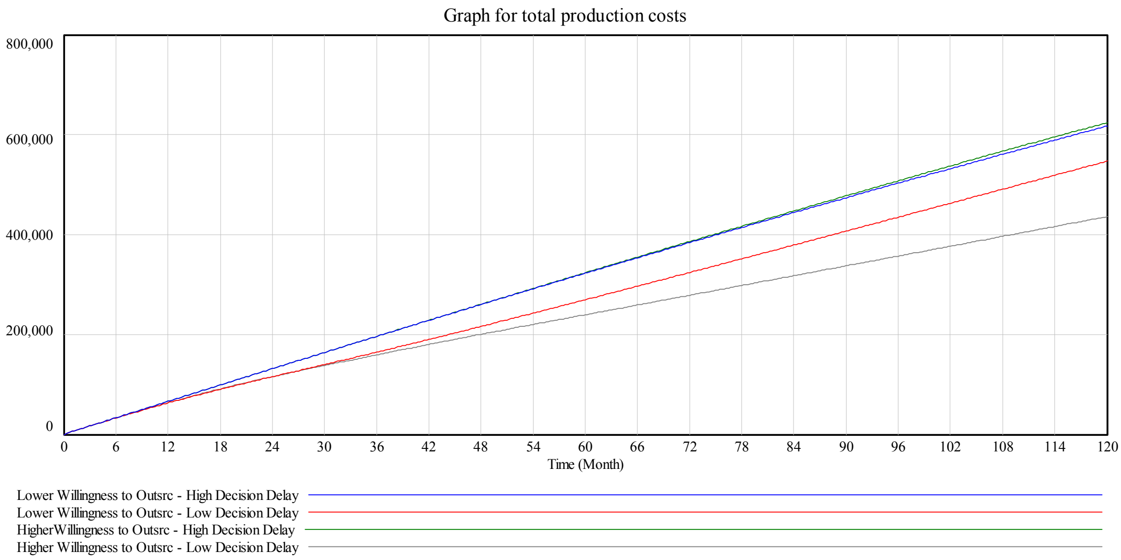


Image 4

The first strategy (blue) has a lower willingness to outsource part of the production and design modular software and has high decision delays. The second strategy (red) also has a low willingness to outsource but with short decision delays.

The third strategy (green) has a higher tendency to plan and design modular software that can be more easily partially outsourced for development and high decision delays.

The last strategy (gray) has a higher willingness to outsource but short delays for deciding to implement the strategy and for implementing it.

As can be seen from Image 4, Total Production Costs can differ greatly depending on which of the four strategies is adopted by the firm. Long delays or wrong timing can substantially increase the total expense, reducing the benefits of outsourcing part of the production. As the graph shows, the wrong outsourcing strategy could cost up to 50% more than the best strategy, which is the gray one.

The model tells us that designing modular software, but not implementing the outsourcing strategy fast enough, is only a partially effective effort and could result in higher costs. Delays then play an important role in defining which strategy is the best.

However, outsourcing part of the total production should be a strategy adopted with force in order to get the most out of it. The second strategy, which has low delays as well, does not reduce the total costs as much as the fourth one because the firm is not aggressive and convinced enough in learning how to design modular software, so it can be partially developed in outsourcing.

One of the main reasons for which the firm who adopts the fourth strategy has lower total production costs can be explained through Image 5.

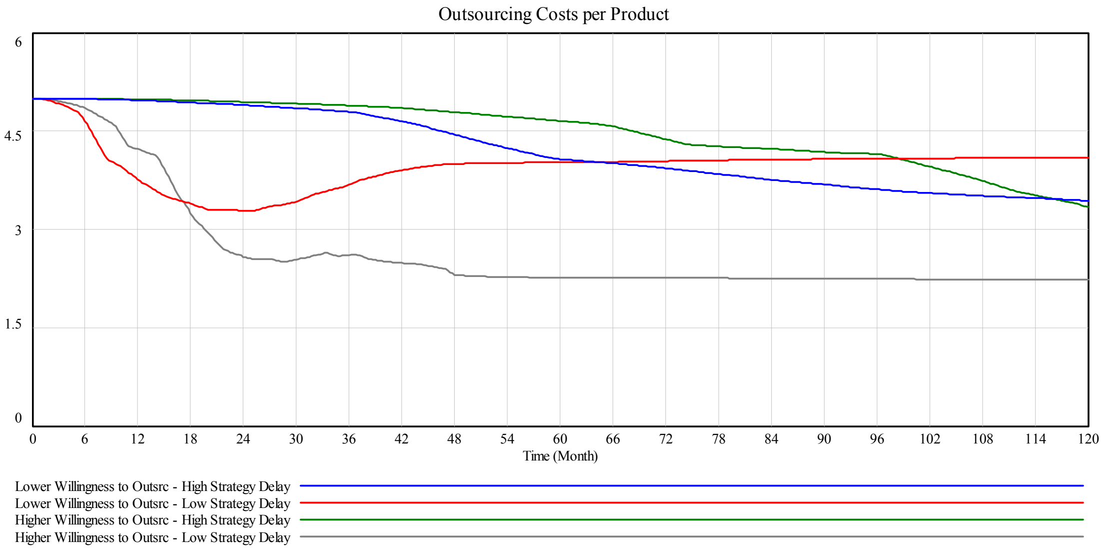


Image 5

In the long run, the gray strategy can substantially reduce the total cost of outsourcing per product, which in the software industry can be measured for example by dollars per lines of code. Compared to the red strategy, using the gray one enables the firm to contrast the supplier's power by reducing the specificity of the software that needs to be purchased by the suppliers. We said that both the red and the gray strategies have short delays. The difference between the two strategies is in the willingness and ability of the firm to design modular software, that can be more easily developed in outsourcing, and the gray one represents a greater ability at doing this.

The red strategy reduces its outsourcing costs per product until it reaches the lowest specificity possible for the module that needs to be outsourced. However, around the 24th month, the firm that adopts the red strategy is not able to simplify further the design of its software and make it more modular. The reinforcing loop connected to the suppliers' power brings the price of outsourcing governance costs up and makes buying that product in outsourcing more expensive.

On the other side, the firm that adopts the gray strategy can reach lower levels of specificity for the software designed. This lower specificity can contrast the suppliers' power and guarantees the firm a lower purchasing cost from the market share for those software modules that will be produced in outsourcing.

Images 6 and 7 show that even if the gray strategy reaches higher values for the stock Fraction of Outsourcing, which illustrates what percentage of the total production is being outsourced, the firm that adopts that strategy has a less expensive flow for outsourcing costs, due to its reduced cost per product. These savings substantially reduce the firm's total production costs in the long run and make the gray strategy the best one to choose.

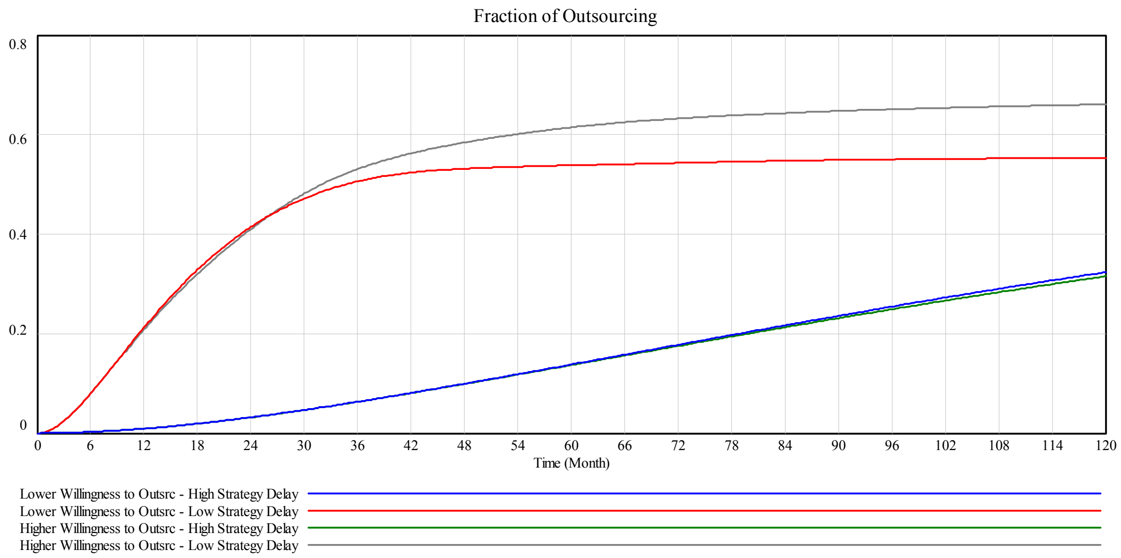


Image 6

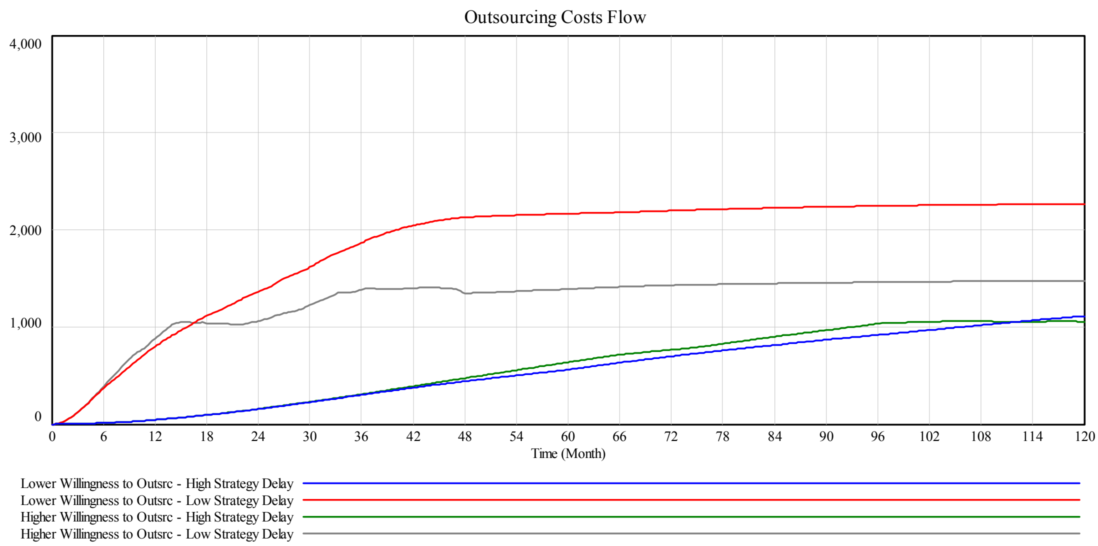


Image 7

CONCLUSION

In this article we used a system dynamics computer simulation to explore the dynamics of outsourcing in the software industry. We grounded our analysis within a typical transaction cost economics theoretical framework and we focused on how the implementation process of an outsourcing decision affects the outcome of the decision. In this respect, we contribute to the analysis of outsourcing by investigating how an outsourcing decision should be implemented and what dynamic consequences the decision may have in the longer term. Thus, the angle we adopted emphasizes the interplay between process and content of strategic decision-making. In this light, we stressed the role of the speed of implementing outsourcing processes and the role of the parallel redesign of productive processes, such as the intervention on the modularity of software produced.

REFERENCES

[ALBSEDSCH03] Alborz S, Seddon, P.B., and Scheepers R., "A Model for Studying IT Outsourcing Relationships", 7th Pacific Asia Conference on Information Systems, Adelaide, July 2003

[ANDAND00] Anderson, E.G., and M.A. Anderson (2000). "Are Your Decisions Today Creating Your Future Competitors? Avoiding the Outsourcing Trap." The Systems Thinker, September 2000: 1-5.

[ANDPAR00] E.G. Anderson, G.G. Parker, "Supply Chain Integration: Putting Humpty-Dumpty Back Together Again", August 2000

[ANDPAR02] Anderson E.J, Parker G.G., "From Buyer to Integrator: The Transformation of the Supply-Chain Manager in the vertically disintegrating firm", Production and Operations Management, Vol 11., N. 1, Spring 2002

[BRUPREPAV01] Brusoni S., Prencipe A., Pavitt K., "Knowledge Specialization, Organizational Coupling, and the Boundaries of the Firm: Why Do Firm Know More Than They Make?", Administrative Science Quarterly, Vol.46, N.4, Dec 2001

[COA37] R.H. Coase, "The Nature of the Firm", Economica, New Series, Vol. 4, No. 16, 1937

[COM34] J.R. Commons, "Institutional Economics", University of Winsconsin Press, 1934

[FUNTIR83] Fudenberg and. Tirole, "Learning-by-Doing and Market Performance", Bell Journal of Economics, 14, 1983

[HEISTU95] Heide J.B., Stump R.L., "Performance Implications of Buyer-Supplier Relationships in Industrial Markets", Journal of Business Research 1995, 32 (57-66)

[HENCLA90] Henderson R., Clark K., "Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms", Administrative Science Quartertly, 35, 9-30

[MAS93], Masten S.E., Transaction Costs, Mistakes, and Performance: Assessing the Importance of Governance, Managerial and Decision Economics, 14:2, 1993

[RIP] Rippa P., "Information sharing in buyer-supplier relationships", University of Federico Napoli II

[SOBSHR98], Sobrero M and Shrader S., "Structuring Interfirm Relationships: A Meta-Analytic Approach", Organization Studies, 19:4, 1998

[SUGRHA], Sullivan, Griswold, Cai, Hallen, "The Structure and Value of Modularity in Software Design"

[ULL00] Ulli Arnold, "New dimensions of outsourcing: a combination of transaction cost economics and the core competencies concept", European Journal of Purchasing & Supply Management, N.6, 2000

[WIL75] O.E. Williamson, "Markets and Hierarchies. Analysis and Antitrust Implications", The Free Press, New York, 1975

[WIL81] O.E. Williamson, "The Economics of Organization: The Transaction Cost Approach", American Journal of Sociology, 1981