ICIS 2002 Proceedings

International Conference on Information Systems (ICIS)

December 2002

Real Options and Software Upgrades: An Economic Analysis

Yonghua Ji
University of Texas at Dallas

Vijay Mookerjee
University of Texas at Dallas

Suresh Radhakrishnan
University of Texas at Dallas

REAL OPTIONS AND SOFTWARE UPGRADES: AN ECONOMIC ANALYSIS

Yonghua Ji
University of Texas at Dallas
Dallas, TX USA
yonghua@student.utdallas.edu

Vijay S. Mookerjee
University of Texas at Dallas
Dallas, TX USA
vijaym@utdallas.edu

Suresh Radhakrishnan
University of Texas at Dallas
Dallas, TX USA
sradhakr@utdallas.edu

Abstract

This work extends earlier work on software upgrades as well as research on real options and IT investment. We consider a two-period model with one software provider who develops and releases a software product to the market. The result shows that the profit from the upgrade policy increases when the market size uncertainty increases. The option value of upgrade is higher when there is more market uncertainty. Also, the value of investing in design effort is more when the development cost is low.

1 INTRODUCTION

The objective of this paper is to gain insights into a software provider's incentives to offer software upgrades using the theory of real options. One of the features of software that is fundamentally different from commodity products is that the option of designing and offering upgrades for software is relatively easier than for commodity products (see Raghunathan 2000). The software provider can use this feature to enhance profits by planning software design such that upgrades are offered when market uncertainty is sufficiently resolved (Dixit and Pindyck 1993; Trigeorgis 1997). While the uncertainty of market size is the driving force behind real options, we examine how the cost of software development and the cost of designing upgrades interact with the uncertainty of market size to affect the value of real options.[1] Typically, with commodity products, the customer has to buy the new upgraded product as a whole, while with software products, customers can buy incremental upgrades and incorporate them into the existing product. In essence, software providers can use software upgrades as a mechanism that enables them to extract real option value. This work extends earlier work on software upgrades such as Ellison and Fudenberg (2000) as well as research on real options and IT investment such as Benaroch and Kauffman (2000) and Sullivan et al. (1999).

[1] The uncertainty of market size is highlighted in the Intuit example. Intuit's chief executive officer, Scott Cook, estimated the potential market for off-line home finance software in 1985 to be close to a $250,000 in California (see *Forbes*, November 11, 1984); while, the market potential in 1995 is $7 million (*San Francisco Chronicle*, May 25, 1995). Two aspects are to be noted in this example: first, there is a potential of increase in market size for software products, and second, the knowledge of this potential creates an option value, which can be incorporated in software design.

2 THE MODEL

We consider a two-period model with one software provider who develops and releases a software product to the market. At the beginning of the first period, the software provider obtains the information on both x_1, the market size in the first period, and x_2, the additional market size in the second period. Specifically, it knows the value of x_1 and obtains a noisy signal on x_2 with mean μ and standard deviation σ. We assume the provider's noisy signal on x_2 is uniformly distributed on the interval $(\mu - \varsigma, \mu + \varsigma)$ with $\sigma = \varsigma / \sqrt{3}$ and $\mu > \varsigma$. The provider determines the number of features (measured using a standard metric such as function points) offered in the initial product (q_1) and the incremental features in period 2 (q_2). Thus, the product has features q_1 in period 1 and ($q_1 + q_2$) in period 2. In addition, at the beginning of period 1, the provider chooses to exert a product design effort, y that can reduce the cost of future upgrades. However, the design effort y does not reduce the first period development cost. The cost of design effort y is $M + ay^2$ where M is the fixed cost.

Consistent with prior literature (Moorthy and Png 1992; Padmanabhan et al. 1996), the cost of developing software is assumed to be $K_1 + bq_1^2$ in period 1 where b represents how expensive it is to develop the software features and K_1 is the fixed cost. Thus the investment by the provider in the first period (C_1) is made up of the direct development cost and the cost of product design effort, i.e., $C_1 = K_1 + bq_1^2 + M + ay^2$. Similarly, the investment in the second period (C_2) is made up of the direct cost of creating an upgrade, i.e., $C_2 = K_2(1-\beta\, y) + b(q_1 + q_2)^2 - bq_1^2$ if the software provider decides to offer an upgrade and is zero otherwise. The first term, $K_2(1-\beta \text{ y})$ captures the benefit of exerting product design effort y in period 1. Examples of product design effort in period 1 that help decrease the future upgrade costs are: more effort in system analysis and design in the first period in anticipation of future upgrades, better design of modules and functions in the first period, and more detailed documentation.[2]

The customers are homogeneous and buy one unit of the product. The utility per unit per period is given by $Q_i + \alpha X_i$ for period i, where Q_i is the value of software for period i, i.e., $Q_1 = q_1$ and $Q_2 = q_1 + q_2$. The network benefit that is derived is αX_i with X_i denoting the number of customers using the software in period i. The new software is backward compatible with the existing software, in the sense that a user of the old software receives the network benefit from the users of the old software; a user of the new software receives the benefit from both user groups, i.e., $X_1 = x_1$ and $X_2 = x_1 + x_2$. Each customer incurs cost c the first time he starts to use the software in both periods and an additional cost c_u if the customer in period 1 buys the upgrade in the second period. We let product design effort y also reduce the learning cost c_u with respect to upgrades, i.e., $c_u = c_0(1-\gamma\, y)$. We assume that the provider and the customers have the same discount factor δ.

3 ANALYSIS AND RESULTS

The provider charges a price P_1 for the base product, P_2 for the new product, and P_u for the upgrade. The profit for the provider in period 1 is $\pi_1 = P_1 x_1 - C_1$ and in period 2, the expected profit is $E(\pi_2) = E(P_2 x_2 + P_u x_1 - C_2)$ if the provider offers the upgrade. The total profit is $\pi_1 + \delta\, E(\pi_2)$. If the provider offers the same product in period 2, then $P_1 = P_2$ and $P_u = 0$.

[2] We do not consider the cost reducing effect of design effort on the variable cost term, $2bq_1q_2$. While this may be reasonable in some situations, the inclusion of this effect makes the model very complex to analyze. We are currently investigating this issue.

We analyze two cases: (1) the benchmark case where the provider does not provide any upgrades, i.e., $q_2 = 0$, and (2) the upgrade case where the provider offers upgrade, i.e., $q_2 > 0$. And we assume that the customers have the same knowledge about the market size as the provider.[3]

The benchmark case corresponds to the situation in which the software provider commits to offering the same product in two periods and the upgrade case corresponds to the situation in which it offers the upgrade. Further we assume that $c_u < \alpha\mu$ so that it is optimal for the provider to offer the upgrade part (see Ellison and Fudenberg 2000). We proceed by examining the benchmark case.

3.1 Benchmark Case

When the software provider offers no upgrades, i.e., $q_2 = 0$, the total expected utility of the two periods for a customer who buys the product in period 1 is $u_1 = (1+\delta)(q+\alpha\cdot x_1)+\delta\cdot\alpha\cdot\mu - c$.And the new customer's utility in period 2 is $u_2 = q+\alpha(x_1+x_2)-c$.

The provider sets the prices at u_1 and u_2 in periods 1 and 2 to extract all customer surplus. Using the optimal prices, the expected profit for the provider in period 1 is

$$\begin{aligned}\pi(q) &= E\left[((1+\delta)(q+\alpha\cdot x_1)+\delta\cdot\alpha\cdot\mu-c)x_1+\delta(q+\alpha(x_1+x_2)-c)x_2\right]-(K_1+bq^2)\\ &= ((1+\delta)(q+\alpha\cdot x_1)+\delta\cdot\alpha\cdot\mu-c)x_1+\delta\{(q+\alpha(x_1+\mu)-c)\mu+\alpha\sigma^2\}-(K_1+bq^2)\end{aligned}$$

Maximizing the provider's expected profit ($\pi(q)$) with respect to q, we get the optimum solution as $q^* = \frac{1}{2b}((1+\delta)x_1+\delta\mu)$. We next derive the optimum solution of the upgrade case.

3.2 Upgrade Case

In this case, the software provider can offer q_1 in the first period and if the market turns out to be favorable in the second period, that is, more new customers come in, then it can offer the new software with the incremental value of q_2, together with the upgraded part for the old software.

The new customer's utility is the same as before since the new software is compatible with the old software even if the existing customers stick with the old software. By offering the upgrade part at a price p_u, the provider can induce the old customers to upgrade.[4] Following the same track as before, the software provider extracts all of the customer's surplus and his expected profit is given by

[3] The provider's market research information is disseminated to customers through articles in trade journals and advertising.

[4] The value of p_u depends on the existing customer coordination rules. Ellison and Fudenberg (2000) discussed two upgrade selection rules: the "reluctant" and the "eager" rules. p_u in the eager upgrade case is higher than that in the reluctant case. However, the discounted payment by the existing customers is independent of p_u. The reason behind that result is because the higher upgrade prices are offset by lower first-period prices as the period 1 customers already expect the future payment for the upgrade.

$$\pi = \bar{\pi}(q_1) + \delta \cdot V_u - \left(M + ay^2\right)$$

where $V_u = \int_{x_{20}}^{\infty} \left((x_1 + x)q_2 - c_u x_1 - C_2\right) f(x)dx$

where C_2 is the investment at the beginning of the second period. x_{20} is the threshold of the new market size at which the provider will break even by developing and offering the new software. If x_2 turns out to be higher than x_{20}, then the provider will develop the new software. Otherwise, the provider will continue to offer the old software to the new customers. Let $h = (x_1 + x_2)q_2 - c_u x_1 - C_2$.The optimal q_2^* satisfies the first-order condition $\frac{\partial h}{\partial q_2} = 0$ and we have $x_1 + x_2 = 2b(q_1 + q_2^*)$. It can be verified that the second-order condition is also satisfied. It follows that

$$h = \frac{(x_1 + x_2)^2}{4b} - q_1(x_1 + x_2) - \left(c_u x_1 + K_2(1 - \beta \cdot y) - bq_1^2\right) \tag{1}$$

Setting $h = 0$, and solving for x_{20}, we have

$$(x_1 + x_{20}) = 2b\left(q_1 + \sqrt{\left((c_u x_1 + K_2(1 - \beta \cdot y)\right)/b}\right) \tag{2}$$

Using the uniform distribution for x_2, (x_2 is between $(\mu - \varsigma, \mu + \varsigma)$), the first order conditions for q_1 and y are:[5]

$$2bq_1 = (1 + \delta)x_1 + \delta\mu + \frac{\delta}{2\varsigma}\left[-\frac{(x_1 + \mu + \varsigma)^2 - (x_1 + x_{20})^2}{2} + 2bq_1(\mu + \varsigma - x_{20})\right] \tag{3}$$

$$2ay = \frac{\delta}{2\varsigma}(\gamma \cdot c_0 x_1 + \beta K_2)(\mu + \varsigma - x_{20}) \tag{4}$$

We present two propositions about the properties of x_{20}, y and q_1, showing how they depend on important parameters such as b and σ.

Proposition 1 (See Appendix for proof)

(1) The threshold market size x_{20} for providing an upgrade increases with b, i.e., $\frac{dx_{20}}{db} > 0$.

(2) The design effort y decreases with b, i.e., $\frac{dy}{db} < 0$.

As b increases, it is more expensive to develop software. The result shows that when it is more expensive to develop software, upgrades will be offered only if the market size has grown considerably. This is in line with our intuition. However, the surprising

[5] The second-order conditions are given in the appendix. They are used to prove propositions 1 and 2.

result of this proposition is $\frac{dy}{db} < 0$. When it is more expensive to develop the software features, then it should become relatively cheaper to put effort y in building better upgrade infrastructure. We would anticipate more effort y. So, $\frac{dy}{db}$ would be positive. But it is not so. This seemingly counter-intuitive result occurs because the threshold of the market size x_{20} increases. Since x_{20} increases, the benefit from the product design effort decreases. This in turn leads to decreased product design effort. Overall, the result shows that the cost of software development and product design effort are complements and not substitutes.

It is also interesting to study the effect of increasing market uncertainty σ on x_{20}, y and q_1. While we can not determine the signs of $\frac{dy}{d\sigma}, \frac{dx_{20}}{d\sigma}, \frac{dq_1}{d\sigma}$ for the general case, we examine cases where the threshold market size is less than the expected additional market size, i.e., $x_{20} < \mu$ and we let the standard deviation vary without affecting the expected additional market size, i.e., σ is independent of μ. We state these as assumption A1:

A1: $x_{20} < \mu$ and σ is independent of μ.

The following proposition provides some observations on the behavior of the product design effort y and the features in period 1, q_1.

Proposition 2 (See Appendix for proof): When A1 is satisfied,

(1) If the design effort decreases with σ, then the threshold market size also decreases with σ, i.e., if $\frac{dy}{d\sigma} < 0$, then $\frac{dx_{20}}{d\sigma} < 0$.

(2) If the threshold market size decreases with σ, then the amount of the initial product features q_1 also decreases with σ, i.e., if $\frac{dx_{20}}{d\sigma} < 0$, then $\frac{dq_1}{d\sigma} < 0$.

Proposition 2 provides some useful insights. Under the condition of $x_{20} < \mu$, when product design effort y decreases as market uncertainty σ increases, then x_{20} will also decrease. The behavior of y and x_{20} with regard to σ is thus different from the behavior with respect to b. We illustrate the results with a numerical example. For purpose of the example, we use the following parameter values and let ς vary over the range of 0 and μ:

$$a = 2;\ c = 40;\ c_0 = 40;\ \alpha = 10^{-4};\ \beta = \lambda = 10^{-3};\ x_1 = 6 \cdot 10^4;\ \mu = 3 \cdot 10^5;\ K_1 = 2 \cdot 10^7;$$
$$K_2 = 4 \cdot 10^6;\ M = 10^6; \delta = 0.8$$

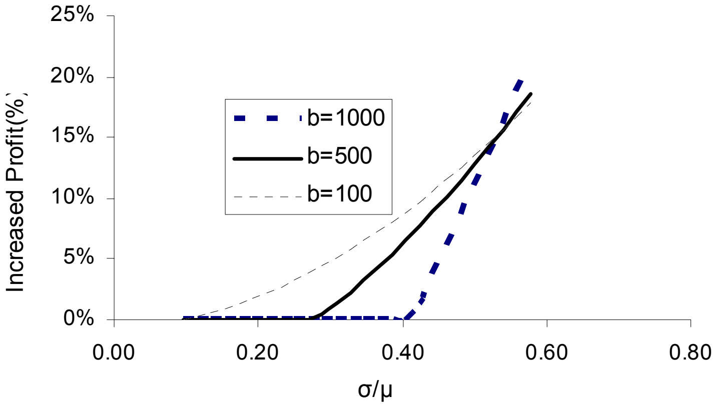

Figure 1. Impact of b and σ on Increased Profit

Figure 1 shows the percentage increased profit of the upgrade case over the benchmark case. The increased profit (%) is calculated by using $(\pi - \overline{\pi})/\overline{\pi} \times 100\ \%$. The figure shows that the profit from the upgrade policy increases when the market size uncertainty increases. The option value of upgrade is higher when there is more market uncertainty, as is generally true with call options (see Dixit and Pindyck 1993). We have also examined the impact of b and σ on the upgrade threshold x_{20} and the initial software features q_1 . The results are the following: (1) For fixed σ, as b increases, y decreases, and x_{20} increases. (2) For fixed b, as σ increases, we have $\frac{dy}{d\sigma} < 0$ and $\frac{dx_{20}}{d\sigma} < 0$ for b = 100. The first part of Proposition 2 is satisfied, but not the other way around. $\frac{dx_{20}}{d\sigma} < 0$ does not necessarily imply $\frac{dy}{d\sigma} < 0$, for the cases of b = 500 and 1000.

4 CONCLUDING REMARKS

One of the key findings of this study is that under the presence of demand uncertainty, the optimal amount of design effort decreases with development cost. As development cost increases, the threshold level of demand required to offer a profitable upgrade also increases. Thus, the probability of offering the upgrade (and hence recovering the investment on design) reduces when the development cost increases. In most situations, development cost would increase with the complexity of an application. The above finding highlights the importance of including demand uncertainty and application complexity in the economics of software design.

This study is a first-step in the analysis of the economic forces that make software upgrades a mechanism for expropriating real option value. By considering heterogeneous customers, future work will include aspects of the market segmentation theory in addition to the real options theory. Other issues to include are competition among software providers and differences between initial development cost and upgrade cost.

5 REFERENCES

Benaroch, M., and Kauffman, R. J. "Justifying Electronic Banking Network Expansion Using Real Options Analysis," *MIS Quarterly* (24:2), 2000, pp. 197-225.

Dixit, A. K., and Pindyck, R. S. *Real Options Under Uncertainty*. Princeton, NJ: Princeton University Press, 1993.

Ellison, G., and Fudenberg, D. "The Neo-Luddite's Lament: Excessive Upgrades in the Software Industry," *RAND Journal of Economics* (31:2), 2000, pp. 253-272.

Moorthy, K., and Png, I. “Market Segmentation, Cannibalization, and the Time of Product Introduction,” *Management Science* (38:3), 1992, pp. 345-359

Padmanabhan, V., Rajiv, S., and Srinivasan, K. “New Products, Upgrades, and New Releases: A Rational for Sequential Product Introduction,” *Journal of Marketing Research* (34), 1996, pp. 456-472.

Raghunathan, S. “Software Editions: An Application of Segmentation Theory to the Packaged Software Market,” *Journal of Management Information System* (17:1), Summer 2000, pp. 87-113.

Sullivan, K. J., Chalasani, P., Jha, S., and Sazawal, V. “Software Design as an Investment Activity: A Real Options Perspective,” in L. Trigeorgis (ed.), *Real Options and Business Strategy: Applications to Decision Making*. London: Risk Books, 1999.

Trigeorgis, L. *Real Options: Managerial Flexibility and Strategy in Resource Allocation*. Cambridge, MA: MIT Press, 1997.

Appendix

To simplify notations, let’s define the following quantities that will be used extensively:

$$T_1 = \frac{\gamma \cdot c_0 x_1 + \beta K_2}{\sqrt{(c_u x_1 + K_2(1-\beta \cdot y))/b}} > 0, \qquad T_2 = \frac{1}{2a} \cdot \frac{\delta}{2\varsigma}(\gamma \cdot c_0 x_1 + \beta K_2) > 0, \qquad T_3 = \frac{y}{\varsigma(\mu + \varsigma - x_{20})} > 0$$

$$T_4 = \frac{\delta \cdot 2b\sqrt{(c_u x_1 + K_2(1-\beta \cdot y))/b}}{2\varsigma - \delta(\mu + \varsigma - x_{20})} > 0, \qquad T_5 = \frac{2(x_1 - 2bq_1) + \delta(2bq_1 + x_1 + \mu - \varsigma)}{2\varsigma - \delta(\mu + \varsigma - x_{20})},$$

For optimal y and q_1 to maximize the profit, the Hessian matrix should be negative definite. That is,

$$\frac{\partial^2 \pi}{\partial q_1^2} = -\frac{2b}{2\varsigma}(2\varsigma - \delta(\mu + \varsigma - x_{20}))(1 - T_4) < 0, \qquad \frac{\partial^2 \pi}{\partial y^2} = 2a(-1 + T_1 T_2) < 0 \text{ and}$$

$$\frac{\partial^2 \pi}{\partial q_1^2} \cdot \frac{\partial^2 \pi}{\partial y^2} - \left(\frac{\partial^2 \pi}{\partial q_1 \partial y}\right)^2 = -4ab(2\varsigma - \delta(\mu + \varsigma - x_{20}))\Delta > 0.$$

Where $\Delta = -1 + T_1 T_2 + T_4$. We can have $(1 - T_4) > 0$, $\Delta < 0$, and $(-1 + T_1 T_2) < 0$ (A.1).

Proof of Proposition 1:

Take the derivative of equations (2), (3), and (4) with respect to b and solve the obtained linear equations for $\frac{dy}{db}$, $\frac{dx_{20}}{db}$, and $\frac{d(2bq_1)}{db}$, we have: $\frac{dx_{20}}{db} = -\frac{\sqrt{(c_u x_1 + K_2(1-\beta \cdot y))/b}}{\Delta} > 0$, $\frac{dy}{db} = -T_2 \frac{dx_{20}}{db} < 0$,

$$\frac{d(2bq_1)}{db} = -\frac{2b\left(\sqrt{(c_u x_1 + K_2(1-\beta \cdot y))/b}\right)^2}{\Delta(2\varsigma - \delta(\mu + \varsigma - x_{20}))} > 0.$$ **Q. E. D.**

Proof of Proposition 2 (Sketch)

Similarly, take the derivative of equations (2), (3), and (4) with respect to ς and solve the obtained linear equations for $\frac{dy}{d\varsigma}$, $\frac{dx_{20}}{d\varsigma}$, and $\frac{d(2bq_1)}{d\varsigma}$, we have: $\frac{dx_{20}}{d\varsigma} = \frac{-T_5 + T_1 T_3(-\mu + x_{20})}{\Delta}$, $\frac{dy}{d\varsigma} = \frac{T_2 T_5 + T_3(-1+T_4)(-\mu + x_{20})}{\Delta}$, and $\frac{d(2bq_1)}{d\varsigma} = \frac{(-1+T_1 T_2)T_5 + T_1 T_3 T_4(-\mu + x_{20})}{\Delta}$.

Using the equations in (A.1), we can prove Proposition 2 easily. Notice that $\sigma = \varsigma / \sqrt{3}$. **Q. E. D.**