Online Supplement to

How Much to Open, How Fast to Fix and Develop? Impacts of Openness on Software Development and Maintenance

Appendix A: Parameter Estimation

We summarize the parameters and variables used in our model in Table 1. The values of the key parameters of our model can be estimated from historical data of the vendor and the market. We now discuss how each of the key parameters of our model are estimated in practice.

- Sensitivities of software quality to openness (i.e., b and y): The sensitivities of the software quality to openness represent how effective the users of open source community are in improving the quality by finding and fixing defects for the current version or developing the next version of the software. The vendors can estimate the sensitivities based on the effectiveness of the open source community using historical data and conducting an empirical study similar to ours. Our empirical study reveals that the defects are fixed faster in open source software compared to proprietary software. This implies that the effect of opening is large and significant, and thus the value of parameter b will be relatively higher. On the other hand, if the empirical model suggests that the difference in time to resolve defects between open source and proprietary software is minimal or of no significance, the value of parameter b will be lower. Hence, depending on the past data and empirical analyses (similar to ours), the values of b can be estimated by the vendor. Likewise, the value for parameter y can be predicted.

- Sensitivities of software quality of the current and next version to the respective efforts (i.e., m and n): As discussed in Section 2.2, the parameter values m and n depend on the productivity of the vendor's resources and the starting quality level of the software. The vendor can adjust the values of these parameters based on prior knowledge of the quality level of the software. That is, if the vendor considers that the initial quality level is low, the value of m will be lower. Besides, we acknowledge that the parameter values of sensitivities to openness (i.e., b and y) and efforts (i.e., m and n) are relative to each other and the vendor can adjust them accordingly.

- Sensitivity of the development of the next version to maintenance effort (i.e., w): The value of w can be calculated based on the extent of dependence of next version on the current version of the software. The

Symbol	Definition
Parameters	
a	Cost multiplier for managing the collaboration with the open source community
b	Sensitivity of the quality of the current version of the software to openness (Effectiveness of openness on the rate of fixing defects)
B	Maximum budget available to the firm for maintenance and development effort (Overall budget/resource constraint)
c_f	Cost multiplier of the maintenance effort (Costliness of the maintenance effort)
c_d	Cost multiplier of the development effort (Costliness of the development effort)
e	Market competition parameter due to openness (Competitiveness of the market)
h	Value of the next version of the software per unit level of quality (Valuation of the next version of the software)
k	Value of the current version of software per unit level of quality (Valuation of the current version of the software)
m	Sensitivity of the quality of the current version to maintenance effort (Effectiveness of maintenance effort on the current version of the software)
n	Sensitivity of the development of the next version to development effort (Effectiveness of development effort on the next version of the software)
T	End of the planning horizon (Release time of the next version of the software)
w	Sensitivity of the development of the next version to maintenance effort (Effectiveness of maintenance effort on the next version of the software)
y	Sensitivity of the development of the next version to openness (Effectiveness of openness on the next version of the software)
Z	Minimum quality level of the next version of the software
θ_0	Intercept for the demand function of the current version
θ_1	Sensitivity of the demand of the current version to quality of the software
θ_2	Sensitivity of the demand of current version to openness
ρ_0	Intercept for the demand function of the next version
ρ_1	Sensitivity of the demand of the next version to quality of the software
ρ_2	Sensitivity of the demand of the next version to openness
Variables	
$D_q(t)$	Demand function for the current version of the software at time t **(State Variable)**
$D_r(t)$	Demand function for the next version of the software at time t **(State Variable)**
$q(t)$	Quality of the current version of the software at time t **(State Variable)**
$r(t)$	Quality of the next version of the software at time t **(State Variable)**
s	Portion of the software code that is open (Extent of openness) **(Decision Variable)**
$u(t)$	Effort of fixing defects in the current version of the software at time t (Maintenance effort) **(Control Variable)**
$v(t)$	Development effort of the next version of the software at time t **(Control Variable)**
$x(t)$	Expenditure of firm on maintenance effort and the development effort at time t **(State Variable)**

Table 1 List of Parameters and Variables

vendor will be able to estimate this sensitivity (i.e., w) relative to the other sensitivities (i.e., b, y, m, and n).

- Valuation of the software (i.e., k and h): The valuations of the software are typically known to the software vendor. Furthermore, they can be estimated based on past sales data, customer requests, market demand, and industry standards.

- Market competition parameter due to openness (i.e., e): The value of this parameter depends on the perceived loss in valuation of the software that the vendor expects when opening up the software. It may also depend on the complexity and the uniqueness of the software code that can be evaluated by the vendor. In particular, if the vendor perceives the code to be unique to them, then the value of e will be very high (i.e., relative to h and k). If there is no reason to believe that the code will be copied or the variants of the code are already commoditized, e will be relatively low compared to h and k.

- Cost of maintenance and development effort (c_f and c_d): These costs consist of expenses related to labor and other resources required to maintain and/or develop new software. Anecdotal evidence suggests that the operating expenses in the software industry is estimated to be around 50% - 75%. Moreover, these parameters can be easily estimated by the software vendors from their past expense reports. Similarly, the cost of collaboration (i.e., a) can be estimated by the vendor based on the resources allocated to managing the collaboration.

- Sensitivity of demand to openness (i.e., θ_2 and ρ_2): Software vendors can estimate this sensitivity terms based on historical data from similar software vendors who have previously made their software partially or fully open source. If the historical evidence suggests that the open source community embraces the openness by joining the software platform, then these sensitivity parameters will be very high. On the other hand, if the community is not very receptive to openness, i.e., the new users are not keen on using the software even though the software is open, these sensitivity parameters will be relatively very low. Similarly, parameters representing the sensitivity of the demand to quality (i.e., θ_1 and ρ_1) can be estimated by software vendors from their previous interactions with the customers and software users.

Appendix B: Proofs and Additional Propositions

Proofs of Lemmas 1, and 3

The Hamiltonian (Sethi and Thompson 2000) of the optimal control problem is given by :

$$H(u,v,s,\lambda_1,\lambda_2,\lambda_3,q,r,x,t) = \lambda_1(t)(bs+mu(t)) - \lambda_3(t)\left(c_d v(t)^2 + c_f u(t)^2\right) + k\left(\theta_0 + \theta_1 q(t) + \theta_2 s\right)$$
$$+\lambda_2(t)(nv(t)+sy+wu(t)) \tag{10}$$

Control variables $u(t)$ and $v(t)$ that are presented in the lemma are derived from solving the following set of equations:

$$\frac{d\lambda_1}{dt} = -\frac{\partial H}{\partial q} = -k\theta_1, \tag{11}$$
$$\lambda_1(T) = k\theta_1(T-t), \tag{12}$$
$$\frac{d\lambda_2}{dt} = -\frac{\partial H}{\partial r} = 0, \tag{13}$$
$$\lambda_2(T) = \rho_1(h-es), \tag{14}$$
$$\lambda_2(T) = \beta, \tag{15}$$
$$\frac{d\lambda_3}{dt} = -\frac{\partial H}{\partial x} = 0, \tag{16}$$
$$\lambda_3(T) = 1, \tag{17}$$
$$\lambda_3(T) = \gamma, \tag{18}$$
$$\frac{\partial H(t)}{\partial u(t)} = -2c_f u(t)\lambda_3(t) + m\lambda_1(t) + w\lambda_2(t) = 0, \tag{19}$$
$$\frac{\partial H(t)}{\partial v(t)} = n\lambda_2(t) - 2c_d\lambda_3(t)v(t) = 0, \tag{20}$$
$$\frac{\partial^2 H(t)}{\partial u(t)^2} = -2c_f\lambda_3(t), \tag{21}$$
$$\frac{\partial^2 H(t)}{\partial v(t)^2} = -2c_d\lambda_3(t). \tag{22}$$

Since the Hamiltonian is strictly concave in u and v (Equations 21 and 22), the solutions presented in Lemmas 1 and 3 are global maxima. Note that, for Lemma 1, Equation (14) is active, whereas in when customer has specific quality requirements (Lemma 3), i.e., when $r(T) = Z$, Equation (15) is active. Next, when the vendor has sufficient budget to cover the expenditure for the effort levels, Equation (17) is active else Equation (18) is active. The presented results are derived by solving the optimal control problem with the corresponding active constraints.

In scenario 1, i.e., when there is no pre-set quality requirement and there is sufficient budget, substituting Equations (12), (15), and (17) into Equation (19) and solving for $u(t)$, we get: $u(t) = \frac{\rho_1 w(h-es)+\theta_1 km(T-t)}{2c_f}$.

Similarly, substituting Equations (14) and (17) into Equation (20) and solving for $v(t)$, we obtain: $v(t) = \frac{n\rho_1(h-es)}{2c_d}$. Hence the results in Lemma 1.

In a similar way, we derive the effort levels for scenario 2, where customer has pre-set quality requirements, we know that $r(T) = Z$. Substituting Equations (12), (15), and (18) into Equation (19) and solving for $u(t)$, we get: $u(t) = \frac{\theta_1 km(T-t)+\beta w}{2c_f}$. Similarly, substituting Equations (15) and (18) into Equation (20) and solving for $v(t)$, we obtain: $v(t) = \frac{\beta n}{2c_d}$. Thus, by substituting the above derived effort levels ($u(t)$ and $v(t)$) into Equation 3 and solving for β, we get: $\beta = -\frac{c_d\left(4c_f(sTy-Z)+\theta_1 kmT^2 w\right)}{2T\left(w^2c_d+n^2c_f\right)}$. Finally, replacing β in $u(t)$ and $v(t)$ with $-\frac{c_d\left(4c_f(sTy-Z)+\theta_1 kmT^2 w\right)}{2T\left(w^2c_d+n^2c_f\right)}$, we obtain: $u(t) = \frac{\theta_1 kmT\left(w^2c_d(T-2t)+2n^2c_f(T-t)\right)+4wc_dc_f(Z-sTy)}{4Tc_f\left(w^2c_d+n^2c_f\right)}$, and $v(t) = \frac{n\left(4c_f(Z-sTy)-\theta_1 kmT^2 w\right)}{4T\left(w^2c_d+n^2c_f\right)}$. Hence the results in Lemma 3. In a similar way, we can derive the effort levels for scenarios with low budget levels.

The Hamiltonian for our model is given in Equation 10. Observe that the openness decision is done at or before t=0. Hence, the Mangasarian conditions require joint-concavity of the Hamiltonian with respect to all state (i.e., q, r, and x) and control (i.e., u and v) variables. Alternatively, we can show the global optimality of our solutions by using Arrow's sufficiency theorem (Seierstad and Sydsaeter 1977). The Arrow's sufficiency requires us to first compute $\widehat{H} = \arg\max H(q,r,x,u,v,s)$. That is $\widehat{H}$ is the Hamiltonian computed along the optimal control trajectories. Then, we need to show that $\widehat{H}$ is jointly concave in q, r, and x. Observe from Lemma 1 that u^*, v^*, and the co-state variables are independent of the state variables. Furthermore, $\widehat{H}$ is linear in q, r, and x and $\frac{d\widehat{H}}{dqdr} = \frac{d\widehat{H}}{dqdx} = \frac{d\widehat{H}}{dxdr} = 0$. Hence, $\widehat{H}$ is indeed jointly concave in q, r, and x. However since $\widehat{H}$ is not strictly concave (or convex) in q, r, and x, Arrow's test is inconclusive about the uniqueness of the optimal solution. Nevertheless, since the necessary conditions (see Equations 19-22) are satisfied only by a single set of controls, our presented solution is indeed unique and hence global optimal. ■

Proof of Lemmas 2 and 4

By substituting the efforts levels derived in Lemmas 1 and 3, the total net value of the software vendor can be written as a function of s in both scenario 1 (no pre-set quality requirement) and scenario 2 (i.e., $r(T) = Z$). These are, respectively, given by:

$$\begin{aligned}\Pi = {} & \frac{c_d\left(T\left(3\theta_1 km\rho_1 Tw(h-es)+3\rho_1^2w^2(h-es)^2+\theta_1^2k^2m^2T^2\right)-6c_f\left(2\left(as^2T-(h-es)\left(\rho_0+\rho_2 s+\rho_1 sTy\right)-\theta_2 ksT\right)-b\theta_1 ksT^2-2\theta_0 kT\right)\right)}{12c_dc_f} \\ & + \frac{3n^2\rho_1^2Tc_f(h-es)^2}{12c_dc_f}\end{aligned} \tag{23}$$

$$\begin{aligned}\Pi_{r(T)=Z} &= k\left(\frac{\theta_1 T\left(12c_f\left(wc_d(bsTw+m(Z-sTy))+bn^2sTc_f\right)+\theta_1 km^2T^2\left(w^2c_d+4n^2c_f\right)\right)}{24c_f\left(w^2c_d+n^2c_f\right)}+\theta_2 sT+\theta_0 T\right)\\ &- \frac{\theta_1^2k^2m^2T^4\left(w^2c_d+4n^2c_f\right)+48c_dc_f^2(Z-sTy)^2}{48Tc_f\left(w^2c_d+n^2c_f\right)}+(h-es)\left(\rho_0+\rho_2 s+\rho_1 Z\right)-as^2T \end{aligned} \tag{24}$$

Hence, by taking the derivative of these functions with respect to s, we find that the optimal portion of the software that is open source in both the scenarios are given by:

$$s^* = \frac{2b\theta_1 kT^2c_dc_f-4e\rho_0c_dc_f-2eh\rho_1^2Tw^2c_d-e\theta_1 km\rho_1T^2wc_d+4h\rho_2c_dc_f+4h\rho_1Tyc_dc_f+4\theta_2 kTc_dc_f-2ehn^2\rho_1^2Tc_f}{2\left(4aTc_dc_f-e^2\rho_1^2Tw^2c_d+4e\rho_2c_dc_f+4e\rho_1Tyc_dc_f-e^2n^2\rho_1^2Tc_f\right)},$$

$$s^* = \frac{c_d\left(w\left(\theta_1 kT^2(bw-my)+2w(-e(\rho_0+\rho_1 Z)+h\rho_2+\theta_2 kT)\right)+4yZc_f\right)+n^2c_f(kT(b\theta_1T+2\theta_2)-2e(\rho_0+\rho_1Z)+2h\rho_2)}{4c_d\left(w^2(aT+e\rho_2)+Ty^2c_f\right)+4n^2c_f(aT+e\rho_2)}.$$

After substituting these optimal values of s^* into the effort levels presented in Lemmas 1 and 3 and the objective function of the vendor, we derive the reported effort levels in Lemmas 2 and 4. The optimal solution for $r(T)^*$ was not presented in Lemma 1 in the main paper. However, we present it here:

$$r^*(T) = \frac{\begin{array}{r}(T\left(c_fn^2\rho_1\left(T\left(4ah-2e\theta_2k\right)-be\theta_1kT^2+2e\left(e\rho_0+h\rho_2\right)\right)+\right.\\ c_d\left(\theta_1kT^2\left(2amw+2bc_fy+e\rho_1w(my-bw)\right)+4c_fy\left(h\rho_2-e\rho_0\right)+2e\rho_1w^2\left(e\rho_0+h\rho_2\right)+\right.\\ \left.\left.2T\left(w\left(\rho_1w\left(2ah-e\theta_2k\right)+e\theta_1km\rho_2\right)+2c_fy\left(h\rho_1y+\theta_2k\right)\right)\right)\right))\end{array}}{2c_f\left(4c_d\left(T\left(a+e\rho_1y\right)+e\rho_2\right)-e^2n^2\rho_1^2T\right)-2c_de^2\rho_1^2Tw^2}$$

Similarly, the optimal solutions for Π^* in Lemma 1 and Lemma 3 can be obtained by substituting the respective values of optimal s in Equations 23 and 24.

Note that scenario 1, in order to have the second-order condition satisfied, we need to have the following condition: $\frac{1}{2}\left(e\left(\rho_1T\left(e\rho_1\left(\frac{n^2}{c_d}+\frac{w^2}{c_f}\right)-4y\right)-4\rho_2\right)-4aT\right)<0$. Further, since $c_f>0$ and $c_d>0$, we can restate this condition as $c_d\left(e^2\rho_1^2Tw^2-4c_f\left(aT+e\rho_2+e\rho_1Ty\right)\right)+e^2n^2\rho_1^2Tc_f<0$. We use this result later in deriving Propositions 1-5.

However, in scenario 2, the second-order condition in the derivation of s^* is always satisfied. ■

Proofs of Proposition 1 and Corollary 1

From Lemma 2, in order to have the second order condition satisfied, we need $\frac{1}{2}\left(e\left(\rho_1T\left(e\rho_1\left(\frac{n^2}{c_d}+\frac{w^2}{c_f}\right)-4y\right)-4\rho_2\right)-4aT\right)<0$. When this condition is satisfied, the objective function is strictly concave with respect to s. Therefore, when $s^*<0$ and the second order condition holds, the objective function is decreasing in the feasible range and objective function is maximum at $s=0$. By analyzing the conditions for $s^*\leq 0$ and second order condition is less than zero (and also ensuring that the feasibility conditions for e are satisfied), we find that the vendor should keep its software proprietary under the following conditions:

(a) $e > \frac{2c_d c_f(T(b\theta_1 kT + 2h\rho_1 y + 2\theta_2 k) + 2h\rho_2)}{c_d\left(4\rho_0 c_f + \rho_1 Tw(2h\rho_1 w + \theta_1 kmT)\right) + 2hn^2\rho_1^2 Tc_f} \equiv \mathscr{E}$, and $a > \frac{1}{4}e\left(e\rho_1^2\left(\frac{n^2}{c_d} + \frac{w^2}{c_f}\right) - \frac{4\rho_2}{T} - 4\rho_1 y\right) \equiv \mathscr{A}$, or

(b) $e > \frac{2c_d c_f(T(b\theta_1 kT + 2h\rho_1 y + 2\theta_2 k) + 2h\rho_2)}{c_d\left(4\rho_0 c_f + \rho_1 Tw(2h\rho_1 w + \theta_1 kmT)\right) + 2hn^2\rho_1^2 Tc_f} \equiv \mathscr{E}$, and $\rho_2 > \frac{1}{4}T\left(-\frac{4a}{e} + e\rho_1^2\left(\frac{n^2}{c_d} + \frac{w^2}{c_f}\right) - 4\rho_1 y\right) \equiv \zeta$.

When the above conditions are satisfied, the optimal value of the extent of openness (i.e., s^*) is equal to zero. Hence the proposition and corollary. ■

Proof of Proposition 2

If the solution suggests that $s^* > 1$, and the second order condition is satisfied, it implies that the objective function value is at maximum when $s = 1$. Therefore, by analyzing the conditions for the inequality $s^* > 1$, we find that the vendor chooses to make its software fully open source under the following conditions:
$b \geq \frac{2c_f\left(2c_d(T(2a + \rho_1 y(2e - h) - \theta_2 k) + \rho_2(2e - h) + e\rho_0) + en^2\rho_1^2 T(h - e)\right) + c_d e\rho_1 Tw(2\rho_1 w(h - e) + \theta_1 kmT)}{2c_f c_d \theta_1 kT^2}$ and $a > \frac{1}{4}e\left(e\rho_1^2\left(\frac{n^2}{c_d} + \frac{w^2}{c_f}\right) - \frac{4\rho_2}{T} - 4\rho_1 y\right)$.
Hence, Proposition 2. ■

Proof of Corollary 2

Using the results from Lemma 2, in order to have the second-order condition satisfied we need to have the following condition satisfied: $c_d\left(e^2\rho_1^2 Tw^2 - 4c_f\left(aT + e\rho_2 + e\rho_1 Ty\right)\right) + e^2 n^2\rho_1^2 Tc_f < 0$. By simple algebra, we obtain the following condition when the above mentioned second-order condition is not satisfied: $a > \frac{1}{4}e\left(e\rho_1^2\left(\frac{n^2}{c_d} + \frac{w^2}{c_f}\right) - \frac{4\rho_2}{T} - 4\rho_1 y\right) = \mathscr{A}$.

Hence, when the second-order condition is not satisfied (so that the objective function is convex), we should have boundary conditions: either $s = 0$ or $s = 1$. Given that the second-order condition is not satisfied, we first solve the optimization problem in Equation (1) at $s = 0$. We get $\Pi_{s=0} = \frac{3h^2\rho_1^2 T\left(w^2 c_d + n^2 c_f\right) + c_d\left(12c_f(h\rho_0 + \theta_0 kT) + \theta_1^2 k^2 m^2 T^3\right) + 3h\theta_1 km\rho_1 T^2 wc_d}{12c_d c_f}$. Next, we solve the optimization problem in Equation (1) at $s = 1$. We get $\Pi_{s=1} = \frac{c_d\left(T\left(3\theta_1 km\rho_1 Tw(h-e) + 3\rho_1^2 w^2(e-h)^2 + \theta_1^2 k^2 m^2 T^2\right) - 6c_f\left(2(aT + (e-h)(\rho_0 + \rho_2 + \rho_1 Ty) - \theta_2 kT) - b\theta_1 kT^2 - 2\theta_0 kT\right)\right) + 3n^2\rho_1^2 Tc_f(e-h)^2}{12c_d c_f}$.

By comparing the Total Net Values at $s = 0$ and $s = 1$, we obtain:

$$(\Pi_{s=1}) - (\Pi_{s=0}) = \frac{c_d\left(c_f\left(2b\theta_1 kT^2 - 4\left(aT + (e-h)\left(\rho_2 + \rho_1 Ty\right) + e\rho_0 - \theta_2 kT\right)\right) + e\rho_1 Tw\left(\rho_1 w(e - 2h) - \theta_1 kmT\right)\right) + en^2\rho_1^2 Tc_f(e - 2h)}{4c_d c_f}.$$

This expression is greater than zero only when:

$$a < \frac{c_d\left(2c_f\left(b\theta_1 kT^2 - 2(e-h)\left(\rho_2+\rho_1 Ty\right) - 2e\rho_0 + 2\theta_2 kT\right) + e\rho_1 Tw\left(\rho_1 w(e-2h) - \theta_1 kmT\right)\right) + en^2\rho_1^2 Tc_f(e-2h)}{4Tc_d c_f} = \bar{\mathscr{A}}.$$

Therefore, when the second-order condition is not satisfied, i.e., $a < \mathscr{A}$ and $a < \bar{\mathscr{A}}$, we have a boundary solution, $s = 1$. Further analysis of these two thresholds (i.e., $\mathscr{A}$ and $\bar{\mathscr{A}}$) suggests that $\bar{\mathscr{A}} < \mathscr{A}$ if and only if $e > \frac{2c_d c_f\left(T\left(b\theta_1 kT + 2h\rho_1 y + 2\theta_2 k\right) + 2h\rho_2\right)}{c_d\left(4\rho_0 c_f + \rho_1 Tw\left(2h\rho_1 w + \theta_1 kmT\right)\right) + 2hn^2\rho_1^2 Tc_f} = \mathscr{E}$. Therefore, when $e > \mathscr{E}$ and $a < \bar{\mathscr{A}}$, we have $s = 1$. Hence part (a) of the corollary.

Next, when $\bar{\mathscr{A}} < a < \mathscr{A}$ and $e > \mathscr{E}$, $s = 0$. Part (a) of Proposition 1 suggests that when the $a > \mathscr{A}$ and $e > \mathscr{E}$ holds, $s = 0$. Hence part (b) of the proposition. ■

Proof of Proposition 3

Analyzing the optimal trajectories of effort levels presented in Lemma 2, we have $u(t)^*$ strictly decreasing in t. Hence, we have $u(0) > u(T)$. Therefore, when $u(T) > v(t)$, the maintenance effort is always greater than the development effort throughout the planning horizon. We find that $u(T) > v(t)$ holds true if and only if $w > \frac{nc_f}{c_d}$. Hence part(a) of the proposition.

Likewise, when $u(0) < v(t)$, the development effort is greater than the maintenance effort throughout the planning horizon. We find that $v(t) > u(0)$ holds true if and only if $h >$

$$\frac{\left[c_f\left(c_d\left(2\theta_1 kT^2\left(4am + e\rho_1(4my - bw)\right) + 4e^2\rho_0\rho_1 w - 4ekT\left(\theta_2\rho_1 w - 2\theta_1 m\rho_2\right)\right) - e^2\theta_1 kmn\rho_1^2 T^2(2n+w)\right) + 2c_f^2 en\rho_1\left(kT\left(b\theta_1 T + 2\theta_2\right) - 2e\rho_0\right) - c_d e^2\theta_1 km\rho_1^2 T^2 w^2\right]}{4c_f\rho_1\left(c_f n - c_d w\right)\left(T\left(2a + e\rho_1 y\right) + e\rho_2\right)}.$$

Hence part(b) of the proposition.

Proof of Proposition 4

Using the results from Lemma 2 and taking the derivative of optimal level of openness and effort levels with respect to k, we get:

$$\frac{dv(t)^*}{dk} = \frac{en\rho_1 T\left(e\theta_1 m\rho_1 Tw - 2c_f\left(b\theta_1 T + 2\theta_2\right)\right)}{4c_d\left(4c_f\left(aT + e\rho_2 + e\rho_1 Ty\right) - e^2\rho_1^2 Tw^2\right) - 4e^2n^2\rho_1^2 Tc_f}, \tag{25}$$

$$\frac{ds^*}{dk} = \frac{Tc_d\left(2c_f\left(b\theta_1 T + 2\theta_2\right) - e\theta_1 m\rho_1 Tw\right)}{c_d\left(8c_f\left(aT + e\rho_2 + e\rho_1 Ty\right) - 2e^2\rho_1^2 Tw^2\right) - 2e^2n^2\rho_1^2 Tc_f}, \quad \textit{and} \tag{26}$$

$$\frac{du(t)^*}{dk} = \frac{\frac{e\rho_1 Twc_d\left(2c_f\left(b\theta_1 T + 2\theta_2\right) - e\theta_1 m\rho_1 Tw\right)}{c_d\left(e^2\rho_1^2 Tw^2 - 4c_f\left(aT + e\rho_2 + e\rho_1 Ty\right)\right) + e^2n^2\rho_1^2 Tc_f} + 2\theta_1 m(T-t)}{4c_f}. \tag{27}$$

From the proof of Lemma 2, in order to have the second-order condition in derivation of s^* to be satisfied, we need to have $c_d\left(e^2\rho_1^2 Tw^2 - 4c_f\left(aT + e\rho_2 + e\rho_1 Ty\right)\right) + e^2n^2\rho_1^2 Tc_f < 0$. Hence, the sign of denominator of Equation (25) is always positive. Now analyzing the inequality in Equation (25), we find that when $b < \frac{em\rho_1 w}{2c_f} - \frac{2\theta_2}{\theta_1 T}$, then $\frac{dv(t)^*}{dk} > 0$. Similarly, the denominator of Equation (26) is always positive, and by simple algebra we find that when $b < \frac{em\rho_1 w}{2c_f} - \frac{2\theta_2}{\theta_1 T}$, then $\frac{ds(t)^*}{dk} < 0$. Next, the sign of the denominator of Equation (27) is always negative. Analyzing this inequality, we find that $\frac{du(t)^*}{dk} > 0$ holds if and only if $b < \frac{c_d\left(4c_f(2\theta_1 m(t-T)(aT+e\rho_2+e\rho_1 Ty)+e\theta_2\rho_1 Tw)+e^2\theta_1 m\rho_1^2 Tw^2(T-2t)\right)+2e^2\theta_1 mn^2\rho_1^2 Tc_f(T-t)}{2e\theta_1\rho_1 T^2 wc_d c_f}$.

Since $\frac{d^2u(t)^*}{dkdt} = -\frac{\theta_1 m}{2c_f} < 0$, we have $\frac{du(t)^*}{dk}$ strictly decreasing in t. Also observe that $\frac{c_d\left(4c_f(2\theta_1 m(t-T)(aT+e\rho_2+e\rho_1 Ty)+e\theta_2\rho_1 Tw)+e^2\theta_1 m\rho_1^2 Tw^2(T-2t)\right)+2e^2\theta_1 mn^2\rho_1^2 Tc_f(T-t)}{2e\theta_1\rho_1 T^2 wc_d c_f}$, evaluated at the upper bound of t, i.e., $t = T$ is $\frac{2\theta_2}{\theta_1 T} - \frac{em\rho_1 w}{2c_f}$. Hence, $\frac{2\theta_2}{\theta_1 T} - \frac{em\rho_1 w}{2c_f} < \frac{c_d\left(4c_f(2\theta_1 m(t-T)(aT+e\rho_2+e\rho_1 Ty)+e\theta_2\rho_1 Tw)+e^2\theta_1 m\rho_1^2 Tw^2(T-2t)\right)+2e^2\theta_1 mn^2\rho_1^2 Tc_f(T-t)}{2e\theta_1\rho_1 T^2 wc_d c_f}$ for any $t < T$. Combining this finding with the above threshold values, we can write the behavior of the effort levels and the extent of openness with respect to k as presented in the proposition. ■

Proof of Proposition 5

Using the results from Lemma 2 and checking for sensitivity with respect to y, we obtain:

$$\frac{ds^*}{dy} = \frac{2\rho_1 Tc_d c_f\left(c_d\left(c_f\left(4ahT-2ekT(b\theta_1 T+2\theta_2)+4e^2\rho_0\right)+e^2\rho_1 Tw(h\rho_1 w+\theta_1 kmT)\right)+e^2hn^2\rho_1^2 Tc_f\right)}{\left(c_d\left(e^2\rho_1^2 Tw^2-4c_f(aT+e\rho_2+e\rho_1 Ty)\right)+e^2n^2\rho_1^2 Tc_f\right)^2} \lesseqgtr 0,$$
$$\frac{du^*}{dy} = \frac{e\rho_1^2 Twc_d\left(-c_d\left(c_f\left(4ahT-2ekT(b\theta_1 T+2\theta_2)+4e^2\rho_0\right)+e^2\rho_1 Tw(h\rho_1 w+\theta_1 kmT)\right)-e^2hn^2\rho_1^2 Tc_f\right)}{\left(c_d\left(e^2\rho_1^2 Tw^2-4c_f(aT+e\rho_2+e\rho_1 Ty)\right)+e^2n^2\rho_1^2 Tc_f\right)^2} \lesseqgtr 0,$$
$$\frac{dv^*}{dy} = \frac{en\rho_1^2 Tc_f\left(-c_d\left(c_f\left(4ahT-2ekT(b\theta_1 T+2\theta_2)+4e^2\rho_0\right)+e^2\rho_1 Tw(h\rho_1 w+\theta_1 kmT)\right)-e^2hn^2\rho_1^2 Tc_f\right)}{\left(c_d\left(e^2\rho_1^2 Tw^2-4c_f(aT+e\rho_2+e\rho_1 Ty)\right)+e^2n^2\rho_1^2 Tc_f\right)^2} \lesseqgtr 0.$$

Further analyzing the above inequalities, we find that when $a < -\frac{e\left(c_d\left(e\rho_1 Tw(h\rho_1 w+\theta_1 kmT)-2c_f(kT(b\theta_1 T+2\theta_2)-2e\rho_0)\right)+ehn^2\rho_1^2 Tc_f\right)}{4hTc_d c_f}$, then $\frac{ds^*}{dy} < 0$, $\frac{du^*}{dy} > 0$, and $\frac{dv^*}{dy} > 0$. Hence, the results in the proposition. ■

Proof of Proposition 6

Using the results from Lemma 4 and checking for sensitivity with respect to k, we get:

$$\frac{dv(t)^*}{dk} = -\frac{nT\left(\theta_1 T\left(amw+bc_f y\right)+2c_f\theta_2 y+e\theta_1 m\rho_2 w\right)}{4\left(c_f\left(T\left(an^2+c_d y^2\right)+en^2\rho_2\right)+c_d w^2\left(aT+e\rho_2\right)\right)}, \tag{28}$$
$$\frac{ds^*}{dk} = \frac{T\left(c_d w\left(\theta_1 T(bw-my)+2\theta_2 w\right)+c_f n^2\left(b\theta_1 T+2\theta_2\right)\right)}{4c_f\left(T\left(an^2+c_d y^2\right)+en^2\rho_2\right)+4c_d w^2\left(aT+e\rho_2\right)}, \quad and \tag{29}$$
$$\frac{du(t)^*}{dk} = \frac{c_f\left(T\left(\theta_1\left(-\left(2amn^2(t-T)+c_d y\left(T(bw-2my)+2mty\right)\right)\right)-2c_d\theta_2 wy\right)+2e\theta_1 mn^2\rho_2(T-t)\right)-c_d\theta_1 mw^2(2t-T)\left(aT+e\rho_2\right)}{4c_f\left(c_f\left(T\left(an^2+c_d y^2\right)+en^2\rho_2\right)+c_d w^2\left(aT+e\rho_2\right)\right)} \tag{30}$$

From the proof of Lemma 4, we know that the objective function is strictly concave in s. Hence, part(a) of the proposition is trivial. The derivative of optimal level of openness (i.e.,s^*) presented in Lemma 4 with respect to k is given by:

$$\frac{ds^*}{dk} = \frac{T\left(c_d w\left(\theta_1 T(bw-my)+2\theta_2 w\right)+c_f n^2\left(b\theta_1 T+2\theta_2\right)\right)}{4c_f\left(T\left(an^2+c_d y^2\right)+en^2\rho_2\right)+4c_d w^2\left(aT+e\rho_2\right)}.$$

This expression is positive if and only if $b > \frac{c_d mwy}{c_f n^2 + c_d w^2} - \frac{2\theta_2}{\theta_1 T}$. In addition, the derivative of the optimal level of maintenance effort (i.e., u^*) presented in Lemma 4 with respect to k is given by:

$$\frac{du(t)^*}{dk} = \frac{c_f\left(T\left(\theta_1\left(-\left(2amn^2(t-T)+c_d y\left(T(bw-2my)+2mty\right)\right)\right)-2c_d\theta_2 wy\right)+2e\theta_1 mn^2\rho_2(T-t)\right)-c_d\theta_1 mw^2(2t-T)(aT+e\rho_2)}{4c_f\left(c_f\left(T\left(an^2+c_d y^2\right)+en^2\rho_2\right)+c_d w^2(aT+e\rho_2)\right)}.$$

This expression is positive if and only if the following holds:

$$c_f\left(T\left(\theta_1\left(-\left(2amn^2(t-T)+c_d y\left(T(bw-2my)+2mty\right)\right)\right)-2c_d\theta_2 wy\right)+2e\theta_1 mn^2\rho_2(T-t)\right)-c_d\theta_1 mw^2(2t-T)(aT+e\rho_2).$$

In the earlier phases of the project (i.e., $t \to 0$), the above expression reduces to $b < \frac{2c_f\left(\theta_1 mT\left(an^2+c_d y^2\right)-c_d\theta_2 wy+e\theta_1 mn^2\rho_2\right)+c_d\theta_1 mw^2(aT+e\rho_2)}{c_f c_d\theta_1 Twy}$.

Further, it is easy to see that $\frac{2c_f\left(\theta_1 mT\left(an^2+c_d y^2\right)-c_d\theta_2 wy+e\theta_1 mn^2\rho_2\right)+c_d\theta_1 mw^2(aT+e\rho_2)}{c_f c_d\theta_1 Twy} > \frac{c_d mwy}{c_f n^2+c_d w^2} - \frac{2\theta_2}{\theta_1 T}$. This is because:

$$\frac{2c_f\left(\theta_1 mT\left(an^2+c_d y^2\right)-c_d\theta_2 wy+e\theta_1 mn^2\rho_2\right)+c_d\theta_1 mw^2(aT+e\rho_2)}{c_f c_d\theta_1 Twy} - \left(\frac{c_d mwy}{c_f n^2+c_d w^2} - \frac{2\theta_2}{\theta_1 T}\right)$$
$$= \frac{m\left(2c_f n^2+c_d w^2\right)\left(c_f\left(T\left(an^2+c_d y^2\right)+en^2\rho_2\right)+c_d w^2(aT+e\rho_2)\right)}{c_f c_d Twy\left(c_f n^2+c_d w^2\right)} > 0.$$

Hence, parts (b), (c), and (d) of the proposition. ■

Appendix C: Empirical Study

In this section, we investigate whether vulnerabilities in open source software environments are fixed faster than their proprietary counterparts. In the next subsection, we discuss the details of the dataset used in our analysis.

C.1. Data

We collected data on 605 vulnerabilities of both closed source and open source software. The vendors used for this study are: SUN, Apache, Apple, Google, IBM, Microsoft, Mozilla, MySQL, Oracle, PHP, and Ruby-on-Rails (ROR). The information about vulnerabilities are collected from the National Vulnerabilities Database *(https://nvd.nist.gov/)*, Common Vulnerabilities and Exposures Website *(https://cve.mitre.org/)*, and the websites of vendors. The variables used in this analysis are summarized in Table 2.

The dependent variable *"Resolving Days"* indicates the number of days taken by a vendor to resolve a particular vulnerability. It is calculated by subtracting the reported vulnerability date in the National Vulnerabilities Database (NVD) from the date the vulnerability patch is available on the vendor's website. The variable *"Software Type"* defines the type of software associated with the vulnerability (open source or

Variable Name	Type	Values	Statistics
Resolving Days	Numeric	17-1198 days	Mean = 128.97, SD = 113.92
Software Type	Classified	Open Source (0), Closed Source (1)	% Open: 85.62 %Closed Source: 14.38
Severity Level	Numeric	1 to 10	Mean = 7.12, SD = 1.96
Access Complexity	Classified	Low, Medium, High	% Low: 40, % Medium: 56 % High: 4%
Vendor	Classified	SUN, Apache, Apple, Google, IBM, Microsoft, Mozilla, MySQL, Oracle, PHP, Ruby-on-Rails (ROR)	–

Table 2 Data Summary

closed source). The *software type* is the main independent variable in our model, which is converted into a dummy variable for the analysis. The variable *"Severity Level"* measures the severity level of vulnerability (denoted by NVD as "CVSS Scores"). This is defined by a numeric number between 1 and 10. In addition to CVSS Scores, the NVD also provides severity rankings of *"Low," "Medium,"* and *"High."* However, these qualitative ratings are simply mapped from the numeric CVSS scores. Specifically, the severity level for a vulnerability is labeled as follows: (i) "*Low*" if it has a CVSS score of 0.0-3.9, (ii) "*Medium*" if it has a CVSS score of 4.0-6.9, and (iii) "*High*" if it has a CVSS score of 7.0-10.0. Since these two measures are equivalent, we consider only numeric values in our analysis.

The variable *"Access Complexity"* measures the complexity of the attack required to exploit the vulnerability once an attacker gains access to the target system. For example, consider a buffer overflow in an Internet service: once the target system is located, the attacker can launch an exploit at will. Other vulnerabilities, however, may require additional steps in order to be exploited. For example, a vulnerability in an email client is only exploited after the user downloads and opens a tainted attachment. The possible values for this metric are: *High, Medium, and Low.* The *High* value indicates that the specialized access conditions exist. For example, the attacking party must already have elevated privileges or spoof additional systems in addition to the attacking system (e.g., DNS hijacking). The *Medium* value indicates that the access conditions are somewhat specialized. For example, the attacking party is limited to a group of systems or users at some level of authorization, possibly untrusted. Finally, the *Low* value indicates that the specialized access conditions or extenuating circumstances do not exist. In this case, the affected product typically requires access to a wide range of systems and users, possibly anonymous and untrusted (e.g., Internet-facing web or mail server). The attack can be performed manually and requires little skill or additional information gathering.

The variable *"Vendor"* indicates the name of the vendor associated with the software for which the vulnerability is reported. For a given software vendor, there may be different software in our dataset. Further, a given vendor may have both closed source and open source software. Table 3 summarizes the number of observations and types of software for each vendor in our dataset. As can be seen, for most of the software vendors in our dataset, the software type is either open source or closed source. However, two vendors (*Apple* and *Oracle*) have both open source and closed source software in our dataset. The variables *Severity Level, Access Complexity,* and *Vendor* serve as control variables in our regression model.

Name of the Vendor	Number of Observations	Types of Software
Apple	240	Open Source and Closed Source
Google	138	Open Source
PHP	48	Open Source
SUN	45	Open Source
Microsoft	45	Closed Source
Apache	34	Open Source
Mozilla	20	Open Source
Oracle	17	Open Source and Closed Source
Ruby-on-Rails (ROR)	9	Open Source
MySQL	8	Open Source
IBM	1	Closed Source
Total	**605**	

Table 3 Types of Software for Different Vendors

C.2. Model and Results

The dependent variable in our model is *Resolving Days* and the key independent variable is the *Software Type* (open source or closed source). We operationalize *Software Type* as a dummy variable: 0 for open source and 1 for closed source. We also employ several control variables: *Severity, Access Complexity,* and *Vendor.* The basic empirical model we test is:

$$ResolvingDays = \beta_0 + \beta_1 Software\,Type + \beta_2 Severity + \beta_3 Access\,Complexity + \gamma Vendordummy + \epsilon. \quad (31)$$

The variable *Access Complexity* with three categories, Low, Medium and High, is operationalized with two dummy variables: ACH and ACM. When the access complexity is high (resp., medium), ACH (resp., ACM)

is equal to 1. In order to overcome potential vendor-specific effects on the relationship between *Software Type* and *Resolving Days*, we consider several operationalizations of *Vendor* as follows:

(a) To evaluate whether vendors who report vulnerabilities more frequently exhibit a distinct and different relationship, we categorize Frequent-Vendors as ones with more than 30 observations in our dataset. As can be seen in Table 3, the following six vendors are in this category: Apache, Apple, Microsoft, Google, PHP, and SUN. Table 4 presents the regression results for all vendors and for frequent vendors.

(b) We introduce vendor specific dummy variables to capture vendor effects through the intercept term. Table 5 presents the regression results for all vendors and for frequent vendors.

(c) While the above specification does capture vendor specific effects, it does so only through the intercept term. In such a specification, all other variables are assumed to exhibit similar relationships with the dependent variable. This assumption is evaluated through an analysis of one Vendor, Apple, which reported 240 vulnerabilities with both open source and closed source software. The regression results are presented in Table 6.

	All Vendors	**Only Frequent-Vendors**
Main Variable		
Closed dummy	30.50(13.63)**	41.53(14.57)***
Control Variables		
Severity	-14.41(2.49)***	-16.64(2.79)***
ACH dummy	-5.58(24.99)	-17.15(26.60)
ACM dummy	-7.95(9.54)	-18.61(10.33)*
Intercept	231.83(18.95)***	254.61(21.23)***
Model Fit		
R^2(F-value)	0.060(9.48)	0.072(10.60)

Note: Standard errors are in brackets.
***: significant at $p = 0.01$, **: significant at $p = 0.05$, *: significant at $p = 0.10$

Table 4 Regression Results for All Vendors and Frequent Vendors

The results consistently reveal that the software type impacts the time to resolve the vulnerability. Hence, when the software type is closed source, it takes longer to resolve the vulnerability (compared to that for the open source software). A robust low-end estimate of 30 additional days for resolution for closed source software is an important insight from the analysis. The results also show that the coefficient for severity is negative and significant across the different model specifications. Thus, more severe vulnerabilities appear

	All Vendors	Only Frequent-Vendors
Main Variable		
Closed dummy	48.65(17.21)***	53.49(18.65)***
Control Variables		
Severity	-7.12(2.52)***	-7.53(2.78)***
ACH dummy	32.05(23.48)	32.36(25.21)
ACM dummy	0.98(11.06)	1.58(11.84)
VendorApple dummy	2.12(34.40)	-149.17(20.78)***
VendorGoogle dummy	73.09(34.41)**	-77.14(21.41)***
VendorIBM dummy	-61.12(106.39)	-
VendorMicrosoft dummy	4.63(40.37)	-150.27(30.60)***
VendorMozilla dummy	70.93(41.00)*	-
VendorMySQL dummy	45.08(48.91)	-
VendorPHP dummy	63.96(36.18)*	-86.88(23.78)***
VendorOracle dummy	-77.31(41.44)*	-
VendorSun dummy	-71.31(36.83)*	-221.41(25.29)***
VendorApache dummy	151.56(37.60)***	-
Intercept	144.04(37.10)***	297.09(24.03)***
Model Fit		
R^2(F-value)	0.256(14.48)	0.237(18.60)

Note: Standard errors are in brackets.
***: significant at $p = 0.01$, **: significant at $p = 0.05$, *: significant at $p = 0.10$

Table 5 Regression Results with Vendor Specific Dummy Variables

	Coefficients (Standard errors)
Main Variable	
Closed dummy	62.28(18.58)***
Control Variables	
Severity	-12.38(4.12)***
ACH dummy	74.75(50.22)
ACM dummy	-38.36(23.05)*
Intercept	216.68(31.58)***
Model Fit	
R^2(F-value)	0.12(8.04)

Note: Standard errors are in brackets.
***: significant at $p = 0.01$, **: significant at $p = 0.05$, *: significant at $p = 0.10$

Table 6 Regression Results for Apple

to be resolved faster. This result can be explained by the fact that the software vendors usually exert extra effort to resolve more severe vulnerabilities. Finally, interestingly, we find that the dummy variables for access complexity are mostly insignificant (except in two scenarios where it registers significance at $p = 0.10$). This result implies that the complexity of the attack required to exploit the vulnerability has little bearing on the time to resolve the vulnerability.