12-5-2011

The Economic Impact of Public Beta Testing: The Power of Word-of-Mouth

Zhengrui Jiang
Iowa State University, zjiang@iastate.edu

Kevin P. Scheibe
Iowa State University, kscheibe@iastate.edu

Sree Nilakanta
Iowa State University, nilakant@iastate.edu

The Economic Impact of Public Beta Testing: The Power of Word-of-Mouth

Abstract

The advent of the Internet has brought many fundamental changes to the way business is conducted. Among others, a growing number of software firms are relying on public beta testing to improve the quality of their products before release. While the benefits resulting from improved software reliability have been widely recognized, the influences of public beta testers on the diffusion of a new software product have not been documented. Through their word-of-mouth effect, public beta testers can speed up the diffusion of a software product after release, and hence increase the time-discounted revenue per adopter. In this research, we take into consideration both the reliability-side and the diffusion-side of the benefits, and develop methodologies to help firms decide the optimal number of public beta testers and the optimal duration of public beta testing. Numerical results show the firm's profit can increase substantially by taking advantage of the world-of-mouth of public beta testers. This benefit is more significant if firms recruit beta testers from those who can benefit from a software product but cannot afford it.

Keywords

Beta testing, software reliability, word-of-mouth, software diffusion, Bass model

Disciplines

Management Information Systems

Comments

This article is from the *ICIS 2011 Proceedings* (December 5, 2011). Posted with permission.

THE ECONOMIC IMPACT OF PUBLIC BETA TESTING: THE POWER OF WORD-OF-MOUTH

Completed Research Paper

Zhengrui Jiang
College of Business
Iowa State University
2340 Gerdin Business Building
Ames, IA 50011-1350
zjiang@iastate.edu

Kevin P. Scheibe
College of Business
Iowa State University
2340 Gerdin Business Building
Ames, IA 50011-1350
kscheibe@iastate.edu

Sree Nilakanta
College of Business
Iowa State University
2340 Gerdin Business Building
Ames, IA 50011-1350
nilakant@iastate.edu

Abstract

The advent of the Internet has brought many fundamental changes to the way business is conducted. Among others, a growing number of software firms are relying on public beta testing to improve the quality of their products before release. While the benefits resulting from improved software reliability have been widely recognized, the influences of public beta testers on the diffusion of a new software product have not been documented. Through their word-of-mouth effect, public beta testers can speed up the diffusion of a software product after release, and hence increase the time-discounted revenue per adopter. In this research, we take into consideration both the reliability-side and the diffusion-side of the benefits, and develop methodologies to help firms decide the optimal number of public beta testers and the optimal duration of public beta testing. Numerical results show the firm's profit can increase substantially by taking advantage of the world-of-mouth of public beta testers. This benefit is more significant if firms recruit beta testers from those who can benefit from a software product but cannot afford it.

Keywords: Beta testing, software reliability, word-of-mouth, software diffusion, Bass model

1. Introduction

Beta testing is the last phase of a software development lifecycle before commercial release. This stage typically begins after alpha testing is completed and all known major issues are resolved. During beta testing, a pre-release version of software is given to a group of users to test in "real world" environments; once detected, bugs are reported back to the development team. Beta testing contributes to quality assurance in several ways. First, beta testers' systems environments can vary in both hardware and software — much more so than the testing environment in a lab. Since the actual users' environments are often difficult or impossible to duplicate in a lab environment, beta testers can help identify defects or compatibilities issues that cannot be detected in testing labs. Second, beta testers are not part of the development team, and thus are likely to be less biased. Third, beta testing helps determine whether the software meets the requirements in terms of functionality and user expectation. The feedback from beta testers can be used to test feature design and help improve marketing and customer support after release.

In traditional beta testing, the internally tested software is distributed only to a small number of selected and trained beta testers. The development team and the beta testers maintain close communication throughout the beta testing process. More recently, with the explosive growth of the Internet, an increasing number of software firms are relying on the Internet to recruit beta testers. To differentiate it from the traditional beta testing practices, this new Internet-enable beta testing practice is sometimes referred to as *public beta testing*.

1.1 Public beta testing

Public beta testing, sometimes also referred to as open beta, is a form of nontraditional beta testing where the beta software is released to a public site and may be downloaded by anyone interested. As more users gain high speed Internet connectivity, they are able to download and test very large beta software releases. The term "beta" has become a well known and well used phrase among computer users. Some examples of well known public betas are Microsoft's Windows Vista community technology previews, where Microsoft would modify features and functionality of the operating system based upon feedback from the beta testing community. Other very popular betas are Google's Gmail, Google Calendar, and Google News (Fester, 2005). Mobile applications for smart phones and tablets have also influenced the public beta testing phenomenon. For example, services such as Jott (www.jott.com) offered an iPhone app that interfaced with a service to transcribe voice to text. Jott went live in December 2006 as a beta. In August 2008, after attracting 420,000 users, Jott left beta and released a premium service (Arrington, 2008). There are even websites specializing in bringing together developers and beta testers for iPhone apps. For example, iBetaTest.com allows iPhone app developers to publish apps, and testers from around the world may to download the apps and report issues and comments back to the developers. As of May, 2011, iBetaTest.com listed over 10,000 beta testers in 36 countries.

Because of the sheer number of public beta testers for a project, the quality of beta testers is more difficult to control in public beta testing than in traditional beta testing. This disadvantage, though, is overshadowed by the availability of a large number of beta testers in a short period of time at substantially lower cost, thus leading to the popularity of the public beta testing practice. Public beta test has become a cost efficient mechanism in software product development because it involves a large number of end-users in the product development life cycle. Also, feature identification and alternations are enabled sooner. Furthermore, in addition to quality assurance, public beta testing also brings other benefits over its traditional beta testing counterpart. One such additional benefit is the improved awareness of the new product, realized through the word-of-mouth of the public beta testers. Based on their own experience, beta testers can spread the quality/availability and other information about the new product to potential adopters, thus speeding up the diffusion of the product throughout its lifecycle. Because of the time-discount factor, a faster diffusion leads to increased present value per sale. Public beta test has been proven useful not only for branded firms to increase market penetration but also for "Web 2.0" firms (firms that are unknown in the marketplace) to gain recognition and market share (Mehr and Shrimali 2008).

Although the software engineering aspect of beta testing is important, the marketing aspect of the problem should also be considered. However, we have not seen any existing research that examines the market related costs and benefits of public beta testing. Taken into consideration both the reliability-side and the diffusion-side of the benefits, the study attempts to address the following questions: How many public beta testers should be recruited and how long should the public beta testing last? How is the solution affected when the marketing related costs and benefits are ignored? Does the composition of the beta testers affect the optimal beta testing strategy and a firm's profitability? How is the beta testing strategy affected in case the beta version has an expiration date?

1.2 Literature review

Compared to other phases of the software development process, beta testing has received less attention. One reference we have seen is a practitioner's guide on beta testing (Fine, 2002). In his discussion of the costs and benefits associated with beta testing, Fine has focused primarily on the software engineering aspect and proposes various guidelines to improve the efficiency of beta testing. Another study (Wiper and Wilson, 2006) uses Bayesian statistical methods to estimate the failure rate and the number of faults during the beta testing phase of a software project. The authors also apply the proposed model to help determine how long the software should be tested and by how many beta testers. Besides benefiting quality assurance, beta testing can also serve as an effective tool for product promotion (Dolan and Mathews, 1993). However, no methodology is proposed to measure the marketing side of the benefit of beta testing.

This study is also related to the literature on free software, since beta software versions could be interpreted as a form of free software. Some of the existing studies have analyzed network externality related benefits of offering free software (e.g., Haruvy and Prasad 1998, Gallaugher and Wang 1999). Two recent articles have also examined the word-of-mouth of free software adopters, and proposed free offer policies based on the benefit resulting from the word-of-mouth effect (Jiang and Sarkar 2010, Jiang 2010). There is a critical difference between free software offer and public beta testing. The purpose of a free offer is not quality assurance, hence software reliability and the reliability-related costs and benefits are not considered in these prior studies. Beta testing focuses on bug detection and product validation, therefore this study explicitly models reliability growth as well as the related benefits such as the reduction in the cost of failures in the field. Furthermore, the two studies that examine the benefits of free adopters' word-of-mouth effect do not treat the duration of free offer as a decision variable, since free offer is assumed to take a negligible amount of time. In this study, the duration of public beta testing is one of the two critical decision variables.

To understand both the software reliability-side and the diffusion-side of the costs and benefits, this study draws on theories and models from two distinct streams of literature: the software reliability literature in software engineering and the product diffusion literature in marketing. The software reliability literature includes software reliability growth models that capture bug detection patterns over time. Such models can be broadly divided into two categories; error-seeding models and failure rate models (Pham 2000, 2006). The widely used Non-Homogeneous Poisson Process model belongs to the second category. The model by Goel and Okumoto (1979), also known as the G-O model, is the most parsimonious NHPP model and is frequently used in various applications of reliability models (McDaid and Wilson, 2001; Xie and Yang 2003). The G-O model shows that the number of undetected bugs decreases at a decreasing rate over time and the bug detection rate at any given time is always proportional to the expected number of undetected bugs. The product diffusion literature in marketing centers on the seminal work by Bass (1969); the model is well known as the Bass model. The Bass model captures the word-of-mouth effect from earlier adopters on future adopters: the larger the number of existing adopters, the more likely that the remaining potential adopters will also adopt. The model has been shown to be applicable to durable goods, non-durable goods, and information goods (Bass 2004). Following the Bass model, numerous extensions and applications have been developed and tested. Among others, the Bass model has been used to study new product growth, dynamic pricing, product entry strategy, and software piracy (Mahajan et al. 2000).

The rest of the paper is organized as follows. In Section 2, we review the Bass model and discuss the reliability assumptions based on the G-O model. In Sections 3 and 4, we examine the various tradeoffs involved in a public beta testing decision, propose models, and analyze solutions for two different cases, one with all public beta testers able to afford the software and the other with a portion of the beta testers not able to afford the product. The third case, where a beta version has an expiration date, is analyzed in

Section 5. We conclude the paper in Section 6 with discussions on managerial implications and future research directions.

2. The Models

We adopt two well-known models to form the theoretical basis for our solution. The word-of-mouth effect of beta testers on future adopters is modeled based on the Bass diffusion model (Bass 1969). The bug detection pattern is modeled based on the Goel and Okumoto Non-Homogenous Poisson Process model (Goel and Okumoto 1979).

2.1 The Bass Model

The Bass diffusion model (Bass 1969) is one of most cited and most influential models in marketing. The model assumes that the probability that a potential adopter will adopt at a given time t is proportional to the number of existing adopters by time t. In other words, the larger is the number of existing adopters, the more likely that those who have not adopted will also adopt at a given time. The direct and indirect influences of the existing adopters on future adopters are also referred to as the word-of-mouth effect. The Bass model can be represented by

$$\frac{dY(t)}{dt} = [p + \frac{q}{m} Y(t)][m - Y(t)], \tag{1}$$

where p is the *coefficient of innovation*, q is the *coefficient of imitation*, m denotes the total number of potential adopters or market size, and $Y(t)$ represents the cumulative number of adoptions by time t. According to Bass (1969), the cumulative number of adoptions at a given time t is

$$Y(t) = \frac{m(1 - e^{-(p+q)t})}{(q/p)e^{-(p+q)t} + 1}, \tag{2}$$

and the non-cumulative rate of adoption at time t equals

$$S(t) = \frac{dY(t)}{dt} = \frac{m(p+q)^2}{p} \frac{e^{-(p+q)t}}{[(q/p)e^{-(p+q)t} + 1]^2}. \tag{3}$$

Once the three parameters (p, q, and m) are known, the entire diffusion path is determined. In the product diffusion literature, these three parameters are often estimated using historical sales data for existing products. Other things being equal, a larger p leads to a higher initial adoption rate, a higher q suggests a larger influence of existing adopters on future adopters. $S(t)$ and $Y(t)$ are proportional to m at any point in time. Besides explaining the diffusion pattern for existing products, the Bass model can also be used to project the future diffusion pattern of a new product based on sales data for similar products (Bass et al. 2001; Bayus 1993).

2.2 The Reliability Growth Model

We model the bug detection process based on the classic Goel and Okumoto Non-Homogenous Poisson Process model (Goel and Okumoto 1979), also known as the G-O model. The G-O model adopts the following important assumption regarding the rate of bug detection:

Assumption 1. The lifetime of each bug is dependent of others, and instantaneous bug detection rate at a given time is always proportional to the number of uncovered bugs at that time.

Based on Assumption 1, the time it takes to detect each given bug follows an independent and identical exponential distribution. Denoting the failure rate of a bug by b, the life time of each bug following the following distribution:

$$f(t) = be^{-bt},$$

and the probability that a given bug will be found before time t is

$$F(t) = 1 - e^{-bt}.$$

We denote the expected number of undetected bugs at the start of testing by N, and that at the time of release (t) by $u(t)$. Based on the G-O model, we have

$$u(t) = Ne^{-bt}.$$

Assumption 2. Public beta testers evaluate the software independently and their collective bug detection efficiency is proportional to the total number of testers.

This assumption is not included in the G-O model since the amount of testing resources is considered exogenous in that model. We adopt it in this study because the number of public beta testers is a decision variable to be determined. Based on this assumption, once the number of public beta testers is given, their collective bug detection efficiency is known and subsequently the G-O can be applied.

Based on Assumptions 1 and 2, if we denote the bug failure rate due to each public beta tester's testing by λ, and the number of beta testers by X, the bug failure rate as a result of the beta testers' collective effort is

$$b = \lambda X.$$

The number of remaining bugs remaining after time τ thus equals

$$u(\tau) = Ne^{-\lambda X \tau}.$$

3. Case I (High-Valuation Beta Testers Only)

Based on whether a potential adopter's reservation price, i.e., the highest price a buyer is willing to pay, is higher than the sale price of a software product or not, we classify all potential adopters of a software product into two classes. A potential adopter belongs to the *high-valuation* class if his reservation price is equal to or higher than the sale price; a potential adopter belongs to the *low-valuation* class if his reservation price is below the sale price. For instance, a college student who is interested in photo editing may benefit from *Adobe Photoshop Lightroom*. If the student is willing to pay the regular price of the software, she belongs to the group of high-valuation potential adopters; otherwise, she should be classified as a low-valuation potential adopter. When a software firm makes its new product publicly available for beta testing, the economic implications are different depending on the class each public beta tester belongs to. In this section, we analyze *Case I*, where all public beta testers are from the class of high-valuation potential adopters. The more general *Case II*, where a portion of the beta testers are from the class of low-valuation potential adopters, will be examined in the next section. For both Case I and Case II, we assume that the word-of-mouth effect from each beta tester is the same as that from a paid adopter.[1]

While firms provide different incentives to encourage participation in beta testing, most public beta testers are drawn by the promise of a free product or the opportunity to try a product that interests them. For both Case I and Case II, we assume that every public beta tester will receive a complete product free of charge at the end of public beta testing. After software release, the beta testers automatically "turn" into adopters of the new software product they have tested.

As explained in the Introduction, public beta testing can affect both the quality and the diffusion of a new software product. In what follows, we first examine the impact of beta testing on the diffusion of a new software product, which determines the total revenue generated during the demand window, and then assess the impact of beta testing on the software's quality, which affects the cost of software failures during operation/usage.

[1] In case their word-of-mouth influences are different, the model can be extended by introducing a parameter to denote the ratio between the word-of-mouth influences for a public beta testers and a paid adopter. We expect the qualitative results to remain valid after this revision.

3.1 Impact of Public Beta Testing on Software Diffusion

We denote the number of beta testers by X and the duration of beta testing by t . For software products, there typically exists a finite demand window (Cohen et al. 1996), denoted by D. The duration of such demand window is often due to various external factors such as advancements in hardware technology, operating systems, or other competing software applications; therefore, we assume that D is exogenous in this research.

Under Case I, all beta testers belong to the high-valuation class, implying that they would purchase the software even if they were charged the regular sale price. By contributing to beta testing, however, these beta testers receive the final product free of charge. Therefore, under Case I, the firm loses the potential revenue from all beta testers. Clearly, this loss increases with the number of public beta testers employed during beta testing. On the other hand, once the product is released, the word-of-mouth effect from these beta testers can help speed up the diffusion of the new software. Because of the discount factor, the increased speed of diffusion can lead to higher time-discounted revenue per adopter. Therefore, in deciding the optimal number of beta testers, a software firm needs to consider the tradeoffs between the loss of potential revenue and the benefits resulting from the word-of-mouth of the beta testers. This tradeoff can be graphically illustrated using the two diffusion curves shown in Figure 1. The curve on the left shows how the diffusion would proceed if the software were released immediately after lab testing. As shown in this curve, the diffusion of the new software would start from a relatively low rate without the word-of-mouth effect of the public beta testers, and hence it would take some time for the diffusion to reach a preferred rate. In contrast, with beta testing, although the commercial launch of the product is delayed, the diffusion can start from a higher rate once the product is released after beta testing. This effect is shown in the curve on the right. For expositional convenience, we term the scenario with beta testing as *scenario BT* and the hypothetical one without beta testing as *scenario NBT*. Based on the basic premise of the Bass model, we arrive at the following conclusion:

Theorem 1. *Under Case I, the diffusion curve after beta testing perfectly matches the portion of the hypothetical diffusion curve, obtained by assuming that the product is released without beta testing, after time θ, where θ is determined by the number of public beta testers X*:

$$\theta = Y^{-1}(X) = \frac{\ln[(pm + qX)/(pm - pX)]}{p+q}.$$

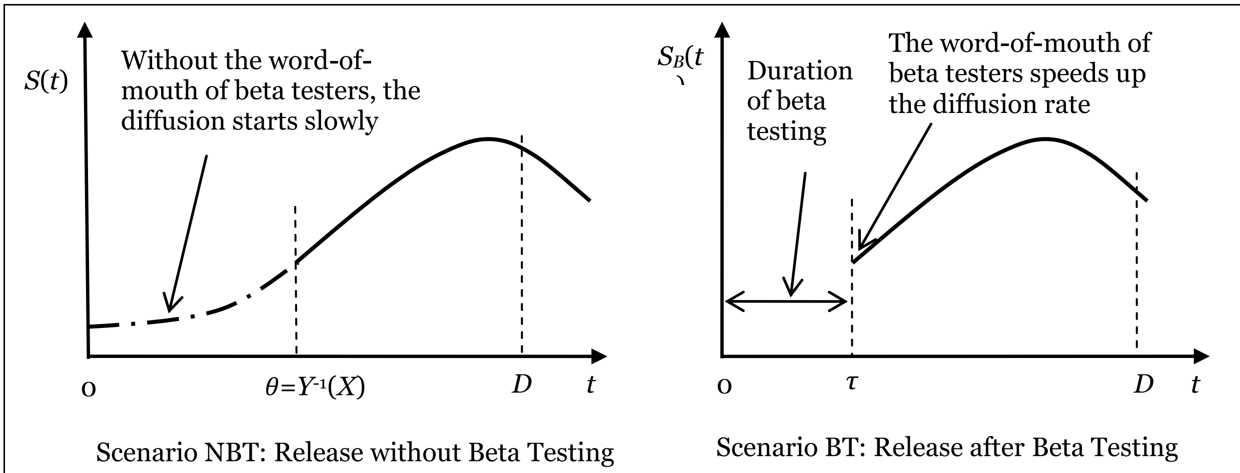

Figure 1. Release with and without Beta Testing

Based on Theorem 1, the diffusion curve for scenario BT matches the solid segment of the curve for scenario NBT. Therefore, we conclude that the word-of-mouth effect of the public beta testers under Case I is equivalent to left-shifting the hypothetical diffusion curve by time θ. By forgoing the potential revenue from time 0 to θ under the hypothetical curve, the firm is able to jump-start the diffusion of the new software product immediately after beta testing. Furthermore, by comparing the distance between θ and D under scenario NBT with the distance between τ and D under scenario BT, it is evident that when the duration of beta testing τ is less than the left-shifting θ, some of the potential adopters, who otherwise would not have the chance to adopt under scenario NBT, will be able to adopt under scenario BT. Therefore, with a relatively shorter beta testing duration, at least a portion of the loss of potential revenue due to the free adoption by public beta testers can be compensated by the purchases made by those who otherwise would not have adopted during the demand window. On the other hand, if $\tau > \theta$, some of the adopters under scenario NBT will not be able to adopt under scenario BT. This is equivalent to shortening the demand window. As a result, the firm loses revenue not only because of the public beta testers, but also because of the shortened sales window. For this scenario to be beneficial, the improvement in software quality as a result of prolonged beta testing must be significant enough to justify the larger loss of potential revenue.

We now derive the present value of the revenue generated after beta testing. Since the two solid curve segments in Figures 2 perfectly matches each other, we can use the hypothetical diffusion rate $S_H(t)$ (after time θ) to obtain the diffusion rate after beta testing. For instance, the diffusion rate at time $(\tau + \varphi)$ for scenario BT equals the diffusion rate at time $(\theta + \varphi)$ scenario NBT. The discount factor, however, is different under the two scenarios unless the duration of beta testing τ equals the amount of left-shifting θ. With a discount rate of r, the discount factor for sales occurring at time $(\tau + \varphi)$ under scenario BT is $e^{-r(\tau+\varphi)}$, while the discount factor for the sales occurring at time $(\theta + \varphi)$ under scenario NBT is $e^{-r(\theta+\varphi)}$. Hence, if we use the diffusion rate $S(t)$ for scenario NBT to represent $S_B(t)$ for scenario BT, the discount factor needs to be appropriately adjusted. Furthermore, the effective sales duration after beta testing is $(D - \tau)$, which also needs to be considered in the model formulation. Assuming a constant price Pr throughout the demand window, the time-discounted revenue after beta testing is represented by

$$R(X,\tau) = Pr \cdot \int_{\theta}^{\theta+D-\tau} S(t)e^{-r(t-\theta+\tau)}dt.$$

3.2 Impact of Public Beta Testing on Software Reliability

We next examine the contributions of the beta testers to the quality of the new software and the resulting benefit. The improvement in quality is a function of both the number of beta testers X and the duration of testing τ. With more public beta testers or a longer testing duration, more bugs are expected to be discovered, leading to a more reliable product. On the other hand, as explained earlier, more high-valuation beta testers result in a larger loss of potential revenue, and a longer testing time delays the firm from reaping the benefit of their investment and may possibly lead to loss of market opportunity. Therefore, these tradeoffs need to be considered in determining the optimal beta testing strategy. To quantify the benefit of improved quality, consistent with the literature (e.g., Dalal and Mallows 1988; Ehrlich et al. 1993), we assume that the total cost of software failures in the field (including the direct cost of fixing the bugs and the indirect costs such as liability cost or loss of goodwill) is a linear function of the number of undetected bugs at the time of release.

Suppose the expected number of bugs just before the start of beta testing by N, and the bug failure rate due to each public beta testers' testing by λ. The average cost per software failure is denoted by c. Based on the reliability assumptions discussed in Section 2.2, the expected number of undetected bugs at the end of beta testing is $u(X,\tau) = Ne^{-\lambda X\tau}$ for X beta testers and a testing duration of τ. The total cost of software failures, therefore, equals

$$L(X,\tau) = cNe^{-\lambda X\tau}.$$

3.3 Problem Formulation and Solution

The profit of the new software product equals the total revenue minus the cost of software failures, i.e., $V(X,\tau) = R(X,\tau) - L(X,\tau)$. The optimal public beta testing problem for Case I can thus be formulated as

$$\begin{aligned}&\underset{X,\tau}{\text{Max}}\ V(X,\tau)=Pr\cdot\int_{\theta}^{\theta+D-\tau}S(t)e^{-r(t-\theta+\tau)}dt-cNe^{-\lambda X\tau},\\&\text{s.t. }\theta=\frac{\ln[(mp+Xq)/(mp-Xp)]}{p+q}.\end{aligned}\tag{4}$$

In order to obtain the best public beta testing solution for a new software product, decision-makers need to project (i) the diffusion curve after release, determined by the three Bass model parameters m, p, and q, (ii) the number of undetected bug at the start of beta testing (N) and the bug failure rate due to a beta tester's testing (λ), and (iii) the expected cost of one software failure duration operation/usage (c). Among these parameters, the diffusion path can be projected based on previous sales data for analogous products (Bayus, 1993; Bass et al., 2001). The parameters N and λ can be estimated by fitting the G-O model to bug detection data for similar products or initial bug detection data for the new product. The expected cost of a software failure can be estimated by domain experts who are familiar with the software's operational environment. This cost should include both the direct cost of identifying and fixing the error and the indirect cost such as liability cost and the loss of goodwill as a result of a software failure.

Analogous to most decision problems of this type, the quality of the solution obtained based on the proposed model depends on how reliable the estimated model parameters are. The formulation of the model, however, does not change with the values of the model parameters. Since no prior study has taken into consideration both software reliability and software diffusion, we are unable to find any diffusion data and bug detection data for the same software project. Instead of using randomly generated values, however, we estimate the parameter values based on data for two different software systems. The Bass model parameters are based on spreadsheet sale data for UK (Givon et al. 1995): p = 0.002, q = 0.648, and m = 1,025K. The expected number of undetected bug at the start of beta testing is estimated to be N = 500 and the bug detection rate per beta tester is assumed to be λ = 0.3. These two parameters are estimated and adjusted based on testing data for a real-time control software (Pham 2006, pp. 144-145). The expected cost of a software failure c, is assumed to be \$50,000. The price of the software (Pr) is set to \$100 and the duration of the demand window (D) is 10 years. Based on these parameter values, we obtained the following optimal solution: X^* = 104K, τ^* = 0.38 year, and the corresponding optimal profit is \$70.98 million.

3.4 What if Beta Testers' Word-of-mouth Effect is Ignored?

In order to understand the impact of the word-of-mouth effect on the solution of public beta testing and a firm's profitability, in this subsection we consider a hypothetical scenario where the beta testers exert no word-of-mouth influence on future adopters. Without the word-of-mouth effect from the beta testers, the left-shifting illustrated in Figure 1 is not applicable. Further, after beta testing, the number of remaining potential adopters decreases from m to (m–X). Based on the Bass model, the diffusion rate at any given point in time is always proportional to the total number of potential adopters. Hence, the problem formulation for this hypothetical scenario is

$$\begin{aligned}&\underset{X,\tau}{\text{Max}}\ V(X,\tau)=Pr\cdot\int_{0}^{D-\tau}\frac{m-X}{m}S(t)e^{-r(t-\theta+\tau)}dt-cNe^{-\lambda X\tau},\\&\text{s.t. }\theta=\frac{\ln[(mp+Xq)/(mp-Xp)]}{p+q}.\end{aligned}\tag{5}$$

Using the same parameter values, we obtain the following optimal solution: X^* = 98K, τ^* = 0.38 year, and the corresponding optimal profit is \$35.9 million. Therefore, even without the word-of-mouth effect of the beta testers, a firm would still recruit public beta testers to help improve the quality of the software. Compared with the scenario where the word-of-mouth from the beta testers is considered, the firm should recruit slightly fewer testers, and the duration of testing should remain about the same; the total profit, however, will drop significantly without the beta testers' word-of-mouth. The drop in profit is due to the fact that the firm cannot increase the speed of diffusion by taking advantage of the word-of-mouth from the public beta testers. This difference can be illustrated by comparing the two diffusion curves with and without the word-of-mouth from the beta testers. From Figure 2, it is evident that without the beta testers' word-of-mouth effect, diffusion will start slowly, and the total sale within the demand window is significantly lower. Therefore, the word-of-mouth effect should be considered when making public beta testing decisions.

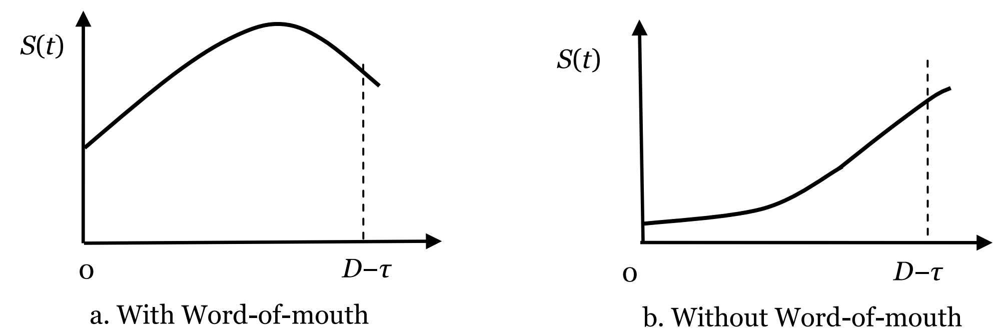

Figure 2. Diffusion Curves with and without the Word-of-mouth of Beta Testers

4. Case II (including low-valuation public beta testers)

Under Case I, we consider the scenario where all beta testers are high-valuation potential adopters. Under a more general Case II, the public beta testers include both high-valuation and low-valuation potential adopters.[2] As defined in the previous section, low-valuation potential adopters have a reservation price lower than the sale price, hence those low-valuation beta testers would not purchase the software if they are charged the regular sale price. Therefore, by recruiting low-valuation potential adopters as beta testers, the firm still benefits from their word-of-mouth; unlike under Case I, however, the firm does not lose any potential revenue from these low-valuation public beta testers. Hence, all else being equal, a larger portion of low-valuation beta testers is more beneficial to a software firm. As the best possible scenario, the firm should recruit beta testers only from the low-valuation potential adopters. However, this ideal scenario is practically impossible because it is difficult to judge whether each potential adopter belongs to the high-valuation class or low-valuation class.

4.1 Problem Formulation and Solution

Suppose the firm recruits $(1+\delta)X$ public beta testers, among which X are from the high-valuation class, and δX are from the low-valuation class. We next examine the impact of the public beta testers on the diffusion of a software product after release. Since the Bass model considers only those potential adopters who can buy the product at the set price, low-valuation potential adopters are not counted in the market size parameter m. To capture the word-of-mouth of these "extra" δX low-valuation adopters, we modify equation (1) for the Bass model as follows:

$$\begin{aligned} &dY(t)/dt = p + (q/m)[Y(t) + \delta X] = p' + (q/m)Y(t), \\ &\text{where } p' = p + (q/m)\delta X. \end{aligned} \tag{6}$$

From (6), we conclude that taking into consideration the word-of-mouth effect of the low-valuation beta testers is equivalent to increasing the coefficient of innovation (p) by $(q/m)\delta X$. With the higher coefficient of innovation, we denote the revised diffusion rate function by

$$S'(t) = \frac{m(p'+q)^2}{p'} \frac{e^{-(p'+q)t}}{[(q/p')e^{-(p'+q)t}+1]^2}. \tag{7}$$

[2] Here we assume that a low-valuation beta bester and a high-valuation beta tester have the bug detection efficiency and the same amount of word-of-mouth influence.

The discounted revenue generated throughout the demand window, becomes

$$R'(X,\tau) = Pr \cdot \int_{\theta}^{\theta+D-\tau} S'(t) e^{-r(t-\theta+\tau)} dt,$$

$$\text{where } \theta = \frac{\ln[(p'm + qX)/(p'm - p'X)]}{p'+q}.$$

We next examine the expected cost of software failures. With the assumption that low-valuation and high-valuation beta testers have the same bug detection efficiency, the expected number of undetected bugs becomes $u'(X,\tau) = Ne^{-\lambda(1+\delta)X\tau}$ at the end of beta testing. The total cost of software failures thus becomes

$$L'(X,\tau) = cNe^{-\lambda(1+\delta)X\tau}.$$

Taking into consideration the revised revenue and cost of software failures, the optimal public beta testing solution for Case II can be obtained based on:

$$\begin{aligned} &\underset{X,\tau}{\text{Max}}\; V(X,\tau) = Pr \cdot \int_{\theta}^{\theta+D-\tau} S'(t) e^{-r(t-\theta+\tau)} dt - cNe^{-\lambda(1+\delta)X\tau}, \\ &\quad \text{s.t. } S'(t) = \frac{m(p'+q)^2}{p'} \frac{e^{-(p'+q)t}}{[(q/p')e^{-(p'+q)t}+1]^2}, \\ &\quad\quad \theta = \frac{\ln[(mp'+Xq)/(mp'-Xp')]}{p'+q}, \\ &\quad\quad p' = p + (q/m)\delta X. \end{aligned} \tag{8}$$

Using the same parameter values for Case I, and let δ=1.0, implying that there are an equal number of low-valuation and high-valuation beta testers, we numerically obtain the optimal solution for Case II: X^* = 87K and τ^* =0.24 year. The optimal profit is \$76.9 million. From this example, we can see that with low-valuation beta testers, the optimal beta testing duration shortens, the optimal number of high-valuation beta testers decreases, while the total number of free adopters (174K) significantly increases. As a result, the optimal profit increases. The increased profit is a result of two factors: first, more beta testers lead to a high speed of diffusion; second, fewer high-valuation beta testers cost the firm less in loss of potential revenue.

4.2 Benefits of Low-valuation Beta Testers

The ratio between the low-valuation and high-valuation public beta testers, δ, may be controlled if the firm is able to identify certain low-valuation potential adopters or if the firm targets a certain segment of the population (e.g., college students) that includes a higher percentage of low-valuation potential adopters. In order to understand the impact of δ, we repeat the numerical analysis for different values of δ. The results[3] are summarized in Figure 3. From this figure, we can see that as the percentage of low-valuation beta testers increases, the optimal testing duration decreases, the optimal number of high-valuation decreases, while the total number of public beta testers increases; and the overall profit increases along the way. From the result, we conclude that a higher δ leads to a number of advantages: (i) the loss of potential revenue from the high-valuation beta testers is lower; (ii) the total number of public beta testers is higher, leading to more bug detection and more word-of-mouth influence on potential adopters after release; and (iii) a shorter beta testing duration increases the present value of each sale. Therefore, a firm should recruit as many as low-valuation beta testers as possible.

[3] For clarity, the optimal duration of public beta testing (τ^*) is converted to days (assuming 250 working days per year) in all figures.

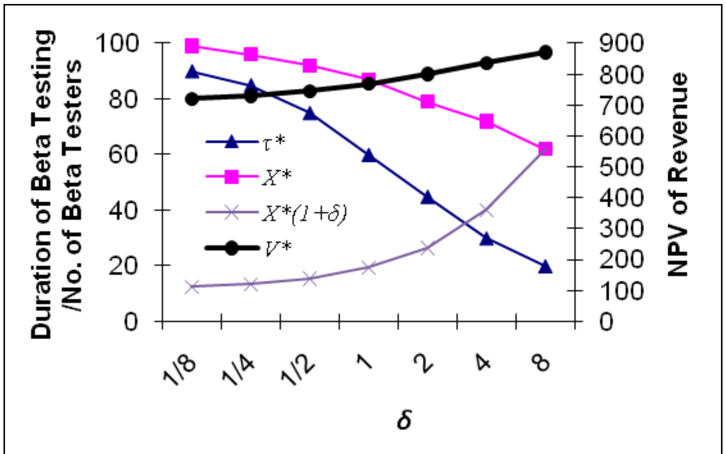

Figure 3. Impact of the Ratio between Low-valuation and High-valuation Beta Testers (δ)

4.3 Sensitivity Analysis

Depending how mission-critical the software product is, the cost of software failures during operation/usage can vary greatly. In order to understand how sensitive the solution is to the cost of software failures, we vary the value of c and record the results as shown in Figure 4. As we can see from the figure, as the expected cost of a software failure increases, it is optimal to recruit slightly more beta testers, and the testing should last longer. The duration of testing increases at a higher pace than the number of beta testers. The net profit decreases because the cost of failure is higher, and the longer testing duration delays the revenue generation.

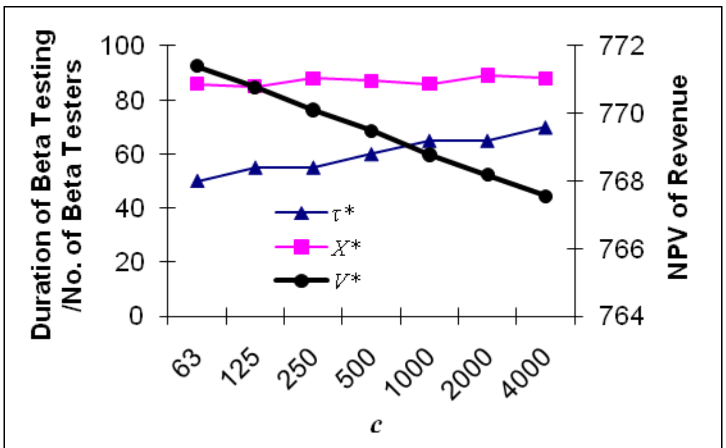

Figure 4. Impact of Cost of A Software Failure (c)

To under the impact of the bug detection rate on the public beta testing solution, we repeat the numerical analysis based on varying values of λ; the results are as shown in Figure 5. We conclude from this figure that if bugs are easier to detect or if the beta testers are more efficient at bug detection, represented by a higher λ, it is optimal to recruit fewer public beta testers and reduce the duration of public beta testing, with the optimal duration of beta testing decreasing at a higher pace than the optimal number of beta testers. Overall, the resulting net profit increases as the bug detection efficiency increases.

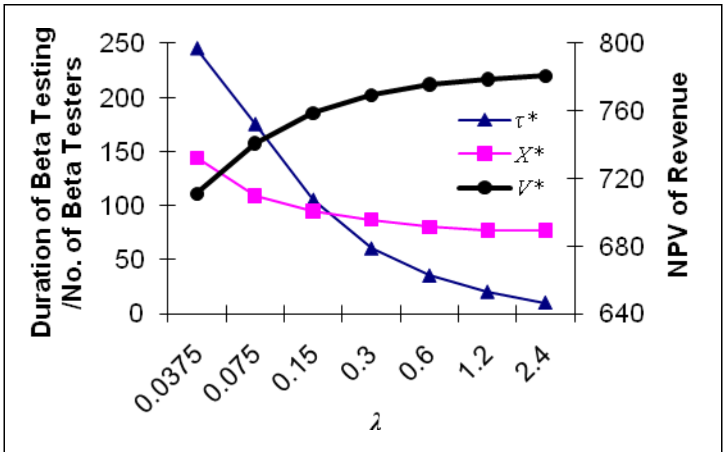

Figure 5. Impact of Bug Detection Efficiency (λ)

In the Bass model, the coefficient of innovation (p) determines the initial diffusion rate after a new product is released, with a high p value implying a higher initial diffusion rate. To understand its impact on the optimal public beta testing solution, we obtain the optimal solutions corresponding to different values of p. The results are shown in Figure 6. From this figure, we can see that as the value of p increases, the optimal number of beta testers decreases, the optimal duration of public beta testing increases, and the resulting net profit increases. We conclude from this result that if the diffusion of a new product can take off relatively quickly by itself, then a firm can recruit fewer testers to reduce of the loss of potential revenue, and the loss in bug detection can be compensated by a slightly longer testing duration.

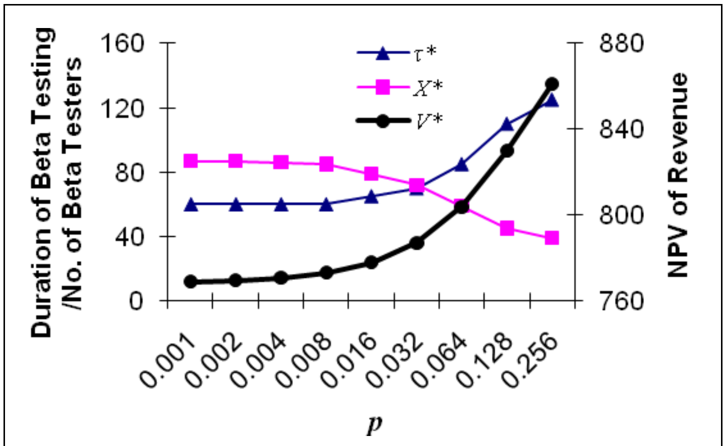

Figure 6. Impact of Coefficient of Innovation (*p*)

We also examine the impact of the coefficient of imitation (q), another important parameter in the Bass model. This parameter measures the amount of influence that existing adopters have on those who have not adopted. A higher q value leads to a faster speed of product diffusion, manifested by a higher adoption rate at the peak and a quicker time to peak. Similar to the other sensitivity analyses, we obtain different optimal solutions corresponding to different values of q. The results are shown in Figure 7. From this figure, we can see that the general impact of q on the optimal solution is similar to that of p. However, the difference is that as the value of q increases, the optimal number of beta testers drops at a much faster rate. This is because q reflects the as the amount of influences from existing adopters on future adopters; as such influences increases, a smaller number of beta testers will be sufficient to spread the word about the product. Once again, the loss of potential revenue can be saved, and the software quality can be assured with longer beta testing duration. The overall profit tends to increase as a result.

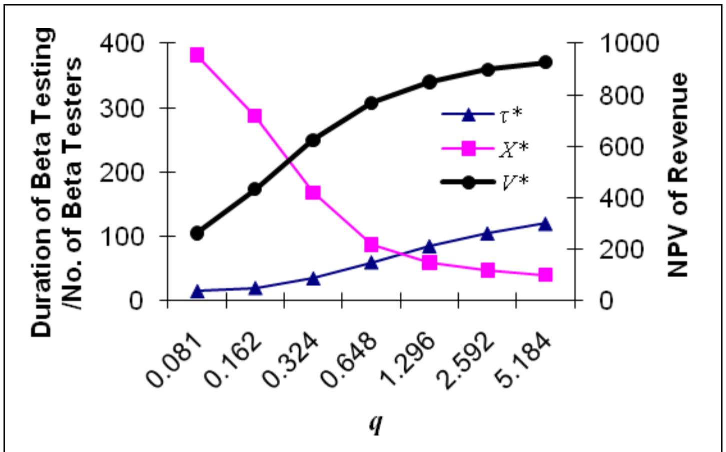

Figure 7. Impact of Coefficient of Imitations (q)

5. Case III (Beta Version with Time Limitation)

Under both Case I and Case II, we assume that all beta testers receive the product for free, and the beta version has no content or time limitation, i.e., the beta version has all the functionality and can be used for a long as the users want. In practice, however, we frequently observe beta software versions coming with a time-limitation. Windows 7 from Microsoft is a good example of such practice. The beta version of the operating system expired on June 1, 2010. After that, public beta testers had to either purchase the commercial version or stop using the operating system. In this section, we examine Case III, under which the beta version available for public beta testing will continue to function for a fixed time period after release. We again assume that the public beta testers include both high-valuation and low-valuation adopters. The low-valuation beta testers will not purchase the commercial software version because their reservation price is lower than the sale price. They will simply stopping using it after the beta version expires. On the other hand, the high-valuation beta testers behave differently. Some may have completed most of the tasks they can perform with the software, and hence decide not to purchase the commercial version. Others may plan to continue to use the software, hence will purchase the commercial version. For those who decide to make the purchase, since they have known the software well enough, we assume that they will purchase the commercial version shortly after the expiration of the beta version. Therefore, the key difference between Case III and Case II is that the firm does not lose the potential revenue from a portion of the high-valuation public beta testers.

We denote the proportion of public beta testers who will also buy the commercial version by γ, and the grace period between the release of the commercial version and the expiration of the beta version by G. The discounted revenue generated by the purchase made by the high-valuation public beta testers thus equals $Pr \cdot \gamma X e^{-r(\tau+G)}$. Therefore, the decision problem for Case III can be formulated as:

$$
\begin{aligned}
&\underset{X,\tau}{\text{Max}}\ V(X,\tau) = Pr\cdot\gamma X e^{-r(\tau+G)} + Pr\cdot\int_{\theta}^{\theta+D-\tau} S'(t)e^{-r(t-\theta+\tau)}dt - cNe^{-\lambda(1+\delta)X\tau},\\
&\text{s.t. } S'(t) = \frac{m(p'+q)^2}{p'}\frac{e^{-(p'+q)t}}{[(q/p')e^{-(p'+q)t}+1]^2},\\
&\theta = \frac{\ln[(mp'+Xq)/(mp'-Xp')]}{p'+q},\\
&p' = p+(q/m)\delta X.
\end{aligned}
\tag{9}
$$

Adopting the same parameter values from the numerical analysis for Case II, and set $\gamma = 0.5$ and $G = 1.0$, we obtain the optimal solution for Case III: $X^* = 151$K, $\tau^* = 0.14$ year, and $V^* = \$82.2$million. In order to further understand how the optimal solution is affected by the percentage of high-valuation public beta testers who will also make the purchase, we vary the value of γ from 0 to 1.0 and summarize the solutions in Figure 8. At $\gamma = 0$, Case III is essentially Case II. From this figure, we can see that as the value of γ increases, the firm should recruit more public beta testers and reduce the testing duration. As expected, the net profit increases monotonically as a larger percentage of high-valuation beta testers are willing to make the purchase after the expiration of the beta version.

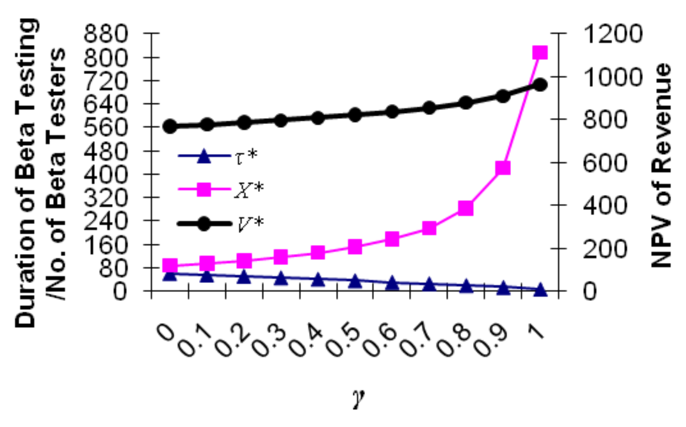

Figure 8. Impact of the Proportion of Beta Testers Who Will Buy (γ)

After comparing problems (8) and (9) and the numerical results, one may be tempted to draw the conclusion that Case III is strictly better than Case II since a firm is able to recover a portion of the loss of revenue from the high-valuation free adopters. In practice, however, the ability for a firm to impose an expiration date for a new product may vary depending on the characteristics and popularity of the product, the firm's reputation, and the underlying market condition. Since public beta testers invest their time in helping improve the product, most of them would expect something in return. Therefore, setting an expiration date on the beta version may affect potential adopters' willingness to participate in beta testing and spread the word about the product. Therefore, for a little-known and low-value product in a competitive environment, offering a beta version without any limitation may be the only realistic way to lure a sufficient number of public beta testers in a reasonable amount of time. Therefore, we should not rush to the conclusion that Case III is always better than Case II. Instead, the public beta testing strategies should be selected based on the market environment and product characteristics; subsequently, one of the proposed models can be used to decide the optimal number of beta testers and the optimal duration of public beta testing.

6. Discussion and Future Research

With the easy accessibility of the Internet, public beta testing has gained tremendous popularity in the software industry. The benefits of public beta testing, however, have not been formally analyzed. In this research, we fill the void by showing that public beta testing not only improves the reliability of a new software product, but also speeds up the diffusion of the product after release. In addition, we show that the benefit of public beta testing can be further enhanced with low-valuation beta testers. The increase in low-valuation beta testers has a significant impact on the profitability of a new software product with a two-fold advantage. First, the testing duration is shortened. Second, high-valuation testers, those who would have paid the full price for the software, are fewer. The combined effect results in potentially higher profit over the life of the product.

Besides improving software reliability and speeding up the diffusion, public beta testing has other practical implications. In a competitive market, the ability to quickly reach a critical mass in customer base can dictate market success or failure. By turning a large number of public beta testers into early adopters, a firm can quickly establish a competitive advantage over its competitors. Furthermore, with a larger base, a firm can later reap other benefits including the sale of newer improved software versions and/or other complimentary products or services.

There are a number of problems that may warrant further study. The cost of adequate mechanisms for handling the increased number of testers is an issue that needs to be factored in future studies. Another issue that is not addressed here is the impact of public beta testing on the quality and marketability of products with successive generations. A third possible direction is to examine the efficacy of public beta testing in speeding up the diffusion of a new product in a competitive market.

References

Bass, F.M. 1969. “A New Product Growth for Model Consumer Durables,” *Management Science* (15:5), pp. 215-227.

Bass, F. M. 2004. “The Bass model: a commentary,” *Management Science* (50:12 Supplement), pp. 1833-1840.

Bass, F.M., Gordon, K., Ferguson, T.L. and Githens, M. L. 2001. “DIRECTV: Forecasting Diffusion of A New Technology Prior To Product Launch,” *Interfaces* (31:3), pp. S83-S93.

Bayus, B.L. 1993. “High-definition Television: Assessing Demand Forecasts for a Next Generation Consumer Durable,” *Management Science* (39:11), pp. 1319-1333.

Cohen, M.A., Eliashberg, J., and Ho, T. 1996. “New Product Development: The Performance and Time-to-Market Tradeoff,” *Management Science* (42:2), pp.173-186.

Dalal, S.R., and Mallows, C. L. 1988. “When should one stop testing software?” *Journal of the American Statistical Association* (83:403), pp. 872-879.

Ehrlich, W., Prasanna, B., Statmpfel, J., and Wu, J. 1993. “Determining the Cost of a Stop-Test Decision,” *IEEE Software* (10:2), pp. 33-42.

Fine, M. R. *Beta testing for better software*, John Wiley & Sons, New York, 2002.

Fester, P. 2005. “A long winding road out of beta,” *ZDNet*, available from http://news.zdnet.com/2100-9588_22-141230.html.

Gallaugher, J. Wang, M., Y. 1999. “Network Effects and the Impact of Free Goods: An Analysis of the Web Server Market,” *International Journal of Electronic Commerce* (3:4), pp. 67-88. Givon, M., Mahajan, V., and Muller, E. 1995. “Software Piracy: Estimation of Lost Sales and the Impact on Software Diffusion,” *Journal of Marketing* (59:1), pp. 29-37.

Goel, A.L. and Okumoto, K. 1979. “Time-Dependent Error-Detection Rate Model for Software and Other Performance Measures,” *IEEE Transactions on Reliability* (R-28:3), 206-211.

Haruvy, E., Prasad, A. 1998. “Optimal Product Strategies in the Presence Of Network Externalities,” *Information Economics and Policy* (10:4), pp. 489-499.

Hu, Q., Saunders, C., and Gebelt, M. 1997. “Research Report: Diffusion of Information Systems Outsourcing: A Reevaluation of Influence Sources,” *Information Systems Research* (8:3), pp. 288-301.

Jiang Z. 2010. “How to give away software with successive versions,” *Decision Support Systems* (49:4), pp. 430-441.

Jiang Z., Sarkar, S. 2010. “Speed Matters: The Role of Free Software Offer in Software Diffusion,” *Journal of Management Information Systems* (26:3), pp. 207-240.

Mahajan, V., E. Muller, Y.Wind. 2000. *New-product diffusion models*. Kluwer, Boston.

McDaid, K. and S. P. Wilson. 2001. “Deciding how long to test software,” *The Statistician* (50:2), pp.117-134.

Mehra, Amit and Shrimali, G. 2008. “Introduction of Software Products and Services Through Public 'Beta' Launches (October 1, 2008),” *NET Institute Working Paper* No. 08-11. Available at SSRN: http://ssrn.com/abstract=1285760.

Pham, H. 2000. *Software Reliability*, Springer, Singapore.

Pham, H. 2006. *System Software Reliability*, Springer, London.

Dolan, R. J., J. M. Matthews. 1993. "Maximizing the utility of customer product testing: Beta test design and management," *Journal of Product Innovation Management* (10:4), pp. 318-330.

Teng, J.T., Grover, V., and Guttler, W. 2002. "Information Technology Innovations: General Diffusion Patterns and its Relationships to Innovation Characteristics," *IEEE Transactions on Engineering Management* (49:1), pp. 13-27.

Wiper, M. P., Wilson, S. P. 2006. "A Bayesian analysis of beta testing," *Test* (15:1), pp.227-255.

Xie, M., B. Yang. 2003. "A Study of the effect of imperfect debugging on software development cost," *IEEE Trans on Software Engineering* (29:5), pp. 471-473.