EXAMINING A BETA TEST

Joseph Wolfe
Experiential Adventures LLC
Jwolfe8125@aol.com

Steven C. Gold
Rochester Institute of Technology
sgold@saunders.rit.edu

Cordelia Norris
Innovative Learning Solutions
cnorris@ilsworld.com

ABSTRACT

It has been strongly recommended that new gaming software undergo a series of tests before its general release. The objective of these tests is to ensure the simulation is appropriate for its intended audience, plays well, possesses the requisite level of fidelity to the system being modeled, and is free from programming errors. This paper first catalogued the design parameters associated with a good beta test. It then compared this ideal against the beta test created for a first-generation online business game released by a major online game publisher. It then examined the actual behaviors and results produced by the study's beta testers to determine the degree the publisher could be confident the game met the criteria of targeted audience propriety, playability, model fidelity and algorithmic accuracy. In this instance, this well-designed beta test could not guarantee the release of error free software.

INTRODUCTION

Software bugs have almost become a modern way of life. All of us have come across accounting errors in our bank account statements, or have been billed for services not rendered. These annoyances are minor and inconsequential compared to some of the world's worst disasters caused by program bugs. Program errors in the Therac-25 radiation therapy machine killed three of six persons given massive radium overdoses (Levenson & Turner, 1993). An onboard program error caused the destruction of the Ariane 5 prototype one minute into its flight on June 4, 1996 at the cost of over $1.0 billion (Dowson, 1997), and a software bug in a Royal Air Force Chinook helicopter's engine control computer caused its crash killing 29 in the process (Roberson, 2002). While these are large-scale events, the phenomena of bug-ridden or poorly tested, business game software is something known to all who have used or created the field's teaching simulation games.

The ability to encode and then release bug-free software is a problem that will not go away. This is especially true when a typical business game's source code, which was once only a few hundred lines, can now encompass thousands of lines. In addition, the cost of detecting bugs is extremely high and the efficacy of code testing efforts is questionable. Given the limited resources possessed by a business game's developer, the bug-finding and debugging task is daunting. Microsoft was still embarrassed by outbreaks of Black, Red and Blue "screens of death" after spending millions of dollars on beta testing their associated operating systems.

A search of the Bernie Keys Library reveals that the mechanics of conducting a valid beta test are not discussed, nor has there been a discussion of what is revealed and not revealed by conducting a beta test. When mentioned at all, game authors either state their game is being beta tested, or that their beta tests were encouraging. Byers and Cannon (2007) discuss how a beta test fits in the game design and development process but do not discuss any details concerning the effective implementation of such a test. This paper will attempt to correct that situation by examining the creation and conduct of an actual beta test, as well as serving to open a debate on the realities of conducting beta tests. To do that this paper will first review the nature of the software testing cycle with a special emphasis on the beta test phase. It will then outline the qualities that should be present to ensure that any beta test results in bug free software. It then presents a case example of the beta test of a newly developed Introduction to Business-level game. This case will highlight the degree to which the test achieved the qualities associated with an ideal beta test, followed by a discussion of the implementation realities of beta testing regardless of the test's design.

THE BETA TESTING PROCESS

Much has been written on the beta testing process for software development. The beta test's goal is to improve the operation and functionality of a software application before its release. Kaner (2006) explains that "software testing" is an empirical evaluation of the quality of the product or service with respect to how it was designed to operate. The testing process will be more effective if the application's developers can articulate and justify how the testing strategy relates to the definition of quality.

The first testing stage before the beta test is referred to as "alpha". This is testing of software that is undergoing in-house testing. An "alpha" turns to a "beta" when software's intended users test the program. Software testing should be done by independent and objective participants. The testing should not be limited to the process of simply finding defects, but should also focus on verifying that the application meets the purpose for which it was designed and programmed.

Most software passes through multiple beta stages and then arrives at "release conditions." A release condition typically requires that all product features have been tested through one or more Beta cycles with no known fatal flaws. A thorough beta test is essential to minimize the risks associated with releasing a software application with significant defects. The final version is commonly referred to as GA ("general availability") or "gold code" for the gold standard expected of released software.

Pan (1999) has identified the key steps in a typical Beta testing process. These steps are the following and are discussed in detail below:

1. Requirements analysis
2. Testing Procedures
3. Reporting Systems
4. Defect Analysis and Re-Testing
5. Closure

REQUIREMENTS ANALYSIS

The first step in a beta test is to develop the requirements list. This list details the software's objectives and expected outcomes. Bach (1999) points out that without such stated requirements no testing is possible because a true beta test compares the software's actual outcomes against its expected outcomes as defined by the product's requirements list.

The requirements list should be based on a clear understanding of the customer's needs. In the case of business game software, this means understanding the targeted student's knowledge and preparation levels and the knowledge domain of the course or business discipline targeted by the game. The development of such a list is no easy task because the ability to recognize problems in a product's design is limited and biased by the designer's understanding or misunderstanding of the nature and purpose of the software's application (Bach, 1999).

For business simulations there are two significant customers-- the student and the instructor or consultant. The requirements list should be developed to meet the needs of both of these customers. The requirements, however, can be general in nature to cover a broad range of objectives, as stated by Bach (1999:114), "There is nothing in the reformulated guidelines that suggests requirements must be made absolutely clear and precise. What these guidelines emphasize is the importance of managing the relationship between risk and a shared understanding of what quality means for your product."

Once the requirements are finalized, the Beta test procedures can be effectively designed.

BETA-TEST PROCEDURES

It is important to formulate and clearly articulate the beta test procedures. Many articles have been written on this subject. For the purposes of this paper we have abstracted what we believe are a beta test's most relevant components. The test process will be more effective if its requirements are specified in terms that communicate the essence of what is desired, along with an idea of risks, benefits, and the relative importance of each requirement (Harmesh, 2009; Kaner, 2006; Shea, 2006).

1. Select qualified participants

A critical component of the Beta test procedure is the selection of its participants or subjects. Kaner (2006) highlighted the importance of selecting independent participants who are managed by objective test administrators. To achieve objectivity in the test's administration there should be no incentives, direct or indirect, for the testers or administrators. It is also important that no participants should be penalized for failure of the Beta test's results. The participants also need to have the background and skills necessary to fulfill the tasks in the Beta test requirements list. This would be best accomplished by randomly drawing the test's participants from the application's target population (Shea, 2006). If random selection cannot be achieved a statistically controlled overt selection process should be employed.

2. Specify test procedures and schedules.

The test procedures should specify how the testers would exercise the test scenarios, including the number of game iterations that will occur and the time schedule involved. It is recommended that the game be run under alternative scenarios if the simulation itself provides flexible applications. When possible the test procedures should cover a wide range of cases, including extreme scenarios and extreme data entry values.

3. Plan and clarify specific roles for testers.

Schedule each tester to focus on a specific test scenario. For critical tests, include more than one tester for each scenario since each tester will approach each task differently.

4. Determine expected results based on "requirements list"

Bach (1999) specifies that all test cases should be traceable to one or more stated requirements, and that these requirements be stated in testable terms. If the software application stores values in a database, pre-determine the expected outcomes so these outcomes can be compared to the application's actual results. For example, in a business simulation, one can compare the decisions made by the students to the expected and actual outcomes with respect to the income statement or balance sheet values and expected ranges.

5. Plan a reward for a job well done.

Shea (2006) points out that a good beta tester shows sincere interest, participation and engagement. To facilitate this goal, Fine (2002) recommends that a special incentive be provided to help promote their involvement. This does not have to be a big reward and items like T-shirts and mugs have been used as effective incentives (Fine, 2002: 42).

REPORTING SYSTEMS

It is important to provide an effective and convenient reporting system for the testers to record defects and other findings. An efficient reporting system will increase the feedback's volume and quality. Several options are suggested, including: designing an online form, a database entry system, an e-mail messaging system, an online discussion board, or any combination of these methods. It is advisable to have testers report problems in real-time, as soon as they are discovered. Reporting in real time is more accurate, timely, and minimizes the probability of not receiving relevant information.

DEFECT ANALYSIS AND RE-TESTING

Compare expected with actual outcomes from the Beta test. Defects reported by the testers must be carefully evaluated for Type I and Type II errors. A Type I error occurs if a tester reports an outcome that is a defect when in fact it is not. A type II error occurs if a defect exists but is not found by the testers. Instituting an effective "test procedure" as described above will help minimize the probability of Type I and Type II errors.

If a defect is found and corrected, based on the severity of the defect and nature of the change in the program, it is advisable to perform a new round of testing. This requires that the complete test procedure be repeated after each round of fixes. It is highly recommended that this step not be skipped. Each time a software program is revised, even when the change seems to be small, it can "break" something else that is even more significant to the program's operation. The only way to be sure a software program is bug-free is to stop the cycles of testing only when no defects are found.

Exhibit 1
Ideal Beta-Test Design Components

QUALIFICATIONS OF PARTICIPANTS
Conducted by independent testers
Conducted by an objective administrator
Played by the application's target population
REQUIREMENTS LIST
Verify the application meets its intended purpose
PROCEDURES
Specify test scenarios and schedules
Clarify the tester's role
Determine the simulation's expected values
Provide incentives for Beta Testers
REPORTING SYSTEM
Provide an effective reporting system for defects and suggestions
DEFECT ANALYSIS
Compare expected outcomes against actual outcomes
CLOSURE
Decision to go or not go to market recognizing the risks associated with making this decision with imperfect information

CLOSURE

A difficult issue for the entire beta testing process is determining when to stop testing and to release the product. It is typically not economically feasible to continue testing until all defects are found and corrected. Yet, the risks could be very high and costly if a product is released with known defects. As pointed out by Yang (1995), testing is a balance between budget considerations, quality, and time. The decision to release a product that does not meet all the design and development features, or has some defects, should be based on a careful analysis of the expected benefits versus the risks and potential costs. The standard economic rule is to stop testing when the expected benefits from continued testing no longer exceeds the expected costs.

A closure meeting between vendors and beta testers is recommended as a final step, with a final report coming from this meeting. This is an effective venue to raise broad questions about the application's learning outcomes, user expectations, and go-to-market or further testing recommendations.

BETA-TEST DESIGN PARAMETERS

Based on a review of the recommended Beta testing process, Exhibit 1 summarizes the key components required for a thorough and complete test. The key components begin with the "qualifications of the participants", including the testers, administrators, and the target population. The "requirements list" must be designed to verify that the application meets its intended purpose. The "procedures" must clearly specify the test scenarios and schedules, the role of the testers, the expected outcomes, and the incentives provided for the testers. The "reporting system" needs to provide an effective method of communication and feedback from the participants. The final steps are a careful "defect analysis" and the "closure decision" to move forward, continue testing or even cancelling the entire release.

HYPOTHESES

The following hypotheses were tested to determine the reliability and quality of the beta test's results as well as the test's components that should result in a useful beta test.

1. All testers will actively participate in playing the game.
2. All testers will be disciplined in their approach to the game.
3. All testers will provide full and complete answers to the study's feedback questions.
4. There will be a high correlation between tester participation and the amount of feedback given.
5. There will be a high correlation between participation and the number of hours billed by the testers.
6. Testers will not commit any Type I errors regarding the game's texts, attributes and programming.
7. Testers will not commit any Type II errors regarding the game's texts, attributes and programming.

METHODOLOGY

Eighteen-business school sophomores at a large southern university served as the study's beta testers. To be a part of the test they had to meet the following criteria:

1. Had not previously played a business game in any form.
2. Dedicate four consecutive weeks to play six decision rounds of a relatively simple online business game.
3. Be one member of a two-member company.
4. Provide endgame feedback via a structured questionnaire.
5. Accept the test's remuneration and billing terms.

Before play began the testers received a copy of the game's 10-page Player's Guide, an instruction sheet on how to access the game at its web address, a copy of the study's questionnaire with instructions for its completion, their license number and game password, and the name and e-mail address of their company's partner. Exhibit 2 presents the six questions the testers were asked to answer. Based on the nature of the questions posed the publisher was primarily conducting a software validation study where the question was "has the right software been written" rather than one of verification where the question is "have we written the software right". The questions also dealt with the software's stability, performance and market/customer reach given its intended audience.

Exhibit 2
Respondent's Questions

Number	Question
1	How suitable is the game for an intro to business class
2	How easy is it to know what you need to do
3	How easy is it to make decisions
4	How easy is it to understand the results
5	How easy is it to understand how to win
6	Did you enjoy playing the game?

The game's turn-in time was by 8:00pm every Monday and Thursday for six rounds of play. The testers received compensation at .25 cents for every spelling and grammar error cited and $5.00 for every math error reported. The testers were also given a staggered hourly budget at $10.00 per hour. This budget allowed more billable hours for the game's opening rounds and fewer for its ending rounds. As an indication of the payout schedule's motivational properties, the state's minimum wage is $7.25 per hour with part-time retail sales clerks earning an average wage of $7.82 in the test's previous year.

The game being tested was *The Global Business Game: Business Basics Edition.* It is a simplified version of *The Global Business Game: World Edition.* Because the game's source code was the same as that used by its mature progenitors, the test was still a Verification test to see if its (1) inherited source code was error-free, (2) author had written the right software for its intended audience and (3) revised its text screens and player support materials in an appropriate manner. The game's breadth and depth was dictated by the topics and tools presented in seven of the field's introduction to business-type textbooks. The following summarizes the games major appearance and playing features:

1. The manufacture of motor scooters in one, two-shift factory.
2. Scooters made from one assembly kit imported from Asia.
3. Two continental markets operating under stable conditions.
4. Financing via stock issues and loans.
5. On-screen call-outs and Help topics.
6. Automated cash flow report
7. A simple, illustrated 24-page step-by-step Player's Guide
8. Excel workbooks and tutorials.
9. Game played via the internet

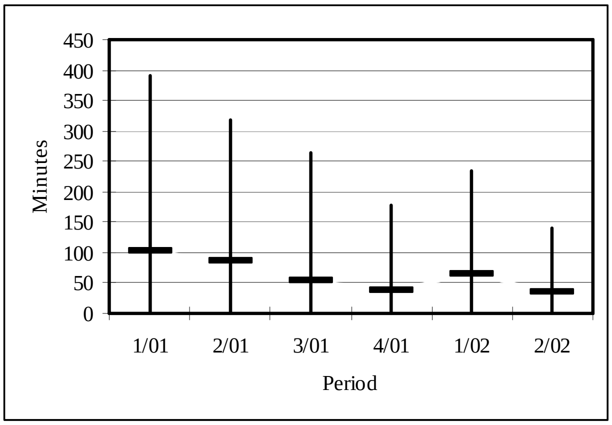
The first two hypotheses were tested using a game administrator feature that records the on-screen time players devote to their game. This feature compiles the start and entry times by player and activity such as print, edit, view, save and submit. The second hypothesis was further examined by retaining all e-mails associated with the game administrator's activities and summarizing those messages that pertained to the game's conduct and orderly processing.

The remaining hypotheses were tested via a content analysis of the responses made to the questions the testers were asked to answer, the information they provided about their experiences and their suggestions for improving the game.

RESULTS

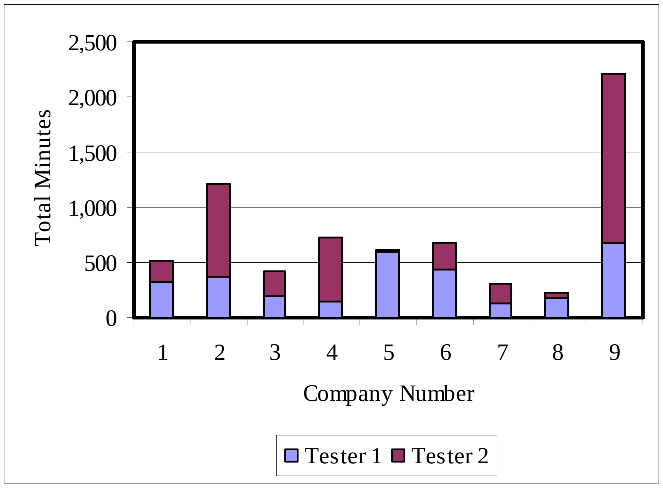
This study's first hypothesis stated all the game's testers would actively participate in playing the game they were evaluating. Exhibit 3 presents a graph of the range of screen time minutes spent by decision period.

Exhibit 3
Screen Time Minutes by Period



This graph indicates that in every period at least one tester spent no time on the period's decision as the range minutes run from zero in every period. The graph also shows that the mean participation rate varied by period, as indicated by the horizontal line intersecting the range line. The greatest time spent on the game was in its first period, with the least amount of time in the last period. Exhibit 4 further indicates the amount of total within-team participation for all periods.

Exhibit 4
Total and Individual Screen Time by Team



The amount of within-team participation equality was the greatest for teams 1 and 3. Firm 5 had the least amount of partnering with 97.5% of company's screen time by one of its members. Other companies, such as teams 2, 4 and 9, one player dominated the other. Based on these two observations Hypothesis 1 is rejected. All players did not actively participate in the game, at least as measured by the amount of time they spent online in an online-based game. In fact, 35.1% of the time certain participants spent less than 5 minutes on that round's decisions.

Exhibit 5
Poor Discipline Incidents

Period	Incident
Pre-Game	A player from Firm1 e-mails the Game Coach that he is not playing the game seriously.
Pre-Game	A player from Firm4 indicates he is unaware there is a Player's Guide to the game even though it was supplied as part of the study's start-up package.
1	Two teams have signed on only one player.
1	Five companies failed to turn-in their first decision set on time.
1	Firm 5 never turns in its decision set. A dummy decision is entered for them by the Game Administrator.
2	The Game Coach sends extensive comments to Firms 5 and 7. None of the teams logon to change their decision sets.
2	The Game Coach suggests to Firm 1 that it look again at its production schedule. The team does not logon to correct this error.
3	Firms 2, 5 and 8 miss the game's turn-in time. Firm 2 never opens its results until after the game's turn-in time.
3	One member of Firm 2 did not know it had a partner in the game. The partner, however, had submitted the team's decision set without the other members' knowledge or approval.
4	Firm 4 submits its decision set 2 ½ days late thereby holding up the entire game.
5	One player on Firm9 states a lack of knowledge that the game had begun and therefore had not been participating in the test.
5	Four firms miss the game's turn-in deadline.
5	Firm 5 submits its decision set 2 ¾ days late.
5	Firm 9 submits its decision set 3 days late.
6	Firm 1 submits its decision set 1 ½ days late.

The study's second hypothesis stated the testers would be disciplined or systematic in their conduct within the test's requirements. All e-mail messages associated with the game were retained. Exhibit 5 presents a log of what could be defined as discipline failures.

Based on the incidents noted above, it could be reasoned the players lacked discipline in a number of areas. They often were unable to submit their decisions on time. This meant they did not begin to work on their next period's decision set early enough even though (1) the test's pacing had been announced in advance and (2) there were from 3-4 days between each decision set. Others did not know they had partners or that the game had begun until after a number of periods had elapsed. Based on these observations the second hypothesis is rejected.

The third hypothesis stated the testers would give full and complete responses to the study's questionnaire. This hypothesis was tested by two methods. Exhibit 6 shows the number of times each question was answered based on the 11 testers who responded and not the 18 testers that *should* have responded.

Exhibit 6
Responses by Question

Question	Count	Percent
How suitable is the game for an intro to business class	7	63.6
How easy is it to know what you need to do	4	36.4
How easy is it to make decisions	6	54.5
How easy is it to understand the results	4	36.4
How easy is it to understand how to win	5	45.5
Did you enjoy playing the game?	3	27.3

Based on these responses the testers answered the questions about the game's suitability for freshmen students and the ease with which decisions could be made. Relatively few answered the questions about enjoying the game or understanding how to win or understanding their results. Exhibit 7 indicates how complete each responding tester's answers were given the questionnaire asked six questions.

Exhibit 7
Answers by Responding Tester

Tester	Answers
1	6/6
2	5/6
3	1/6
4	0/6
5	0/6
6	2/6
7	6/6
8	4/6
9	2/6
10	1/6
11	2/6
Total	2.64/6.00

Taken in total this hypothesis regarding complete responses must be rejected as nine of eighteen testers either did not answer any question or did not respond to the questionnaire. Of those who *did* respond, only three testers answered all questions or nearly all the questions. These partial responses resulted in a 44.0% question response rate. Accordingly, this hypothesis of response completeness was rejected.

It should be noted, however, that some of the testers were very diligent and business-like with the responses they provided. The questionnaire asked them to document all grammatical errors with screen captures, quotes and links to the errors. They responded by presenting 50 screen captures, 113 suggested sentence re-wordings and 35 links to the sources of errant texts.

The fourth hypothesis looked to see if there was a high ratio between the amount of time the testers put into playing the game and the amount of information they provided. It was assumed those who put in the most time viewing and interacting with the game would also have the most to state about the game. This hypothesis was mildly accepted. The correlation between the amounts of screen time devoted the game, and the number of words found in their reports was low but statistically significant with an r-square of 0.182 and a p-value of 0.04 in a one-tail test. This weak correlation means that 81.8 percent of the variation in the size of their reports was associated with other factors.

The fifth hypothesis tested whether the game publisher's dollar payout reflected the number of hours of tester game time and the feedback game play was supposed to produce. It would be assumed that those who spent the greatest amount of game time would provide the greatest amount of feedback. They should also bill the greater number of hours as their reward for their efforts. It has already been determined there is a positive but weak relationship between the amount of time the testers spent playing the game and the size of their reports. Exhibit 8 shows the actual hourly rate the testers were paid based on the amount of time they spent online with the game and the size of their reports. The game pay rate ranged from $0.00 to $20.31 an hour with an average hourly cost of $10.75. The cost per report word is more problematical because eight testers received pay for play even though they did not file a report. If a report was filed, the pay per report word ranged from 0.04 cents to 0.73 cents per word or an average cost of 0.27 cents per word. A regression analysis of the testers who filed a report, after constraining the intercept value to zero, shows that Report Size is positively related to the Pay Rate. It is statistically significant with a P-value less than 5.0% and an adjusted r-square equal to 32.9%. Clearly, some testers were better values for the publisher both in the number of hours they devoted to the game and the amount of words they produced per billed hour.

Exhibit 8
Pay Per Online Game Time and Report Size

Tester	Game	
	Pay Rate	Report Size
1	$11.42	N/R
2	$15.22	N/R
3	$10.06	0.17
4	$8.53	0.10
5	$16.18	0.28
6	$20.31	0.07
7	$10.00	N/R
8	$11.81	0.53
9	$10.00	0.73
10	$0.00	N/R
11	$8.99	0.60
12	$13.79	N/R
13	$12.57	0.04
14	$10.23	N/R
15	$9.84	N/R
16	$10.00	N/R
17	$10.11	0.04
18	$4.45	0.16
Average	$10.75	0.27

This study's last two hypotheses tested the degree the respondents made accurate statements about the game's screen texts, navigational attributes and programming fidelity. In a statistical test, a Type I error is one where it is stated the condition is True when it is actually False. In this study's beta test, it means the testers did not find an error when an error was ultimately found to exist. A Type II error is one where it is stated the condition is False when it is actually True. As an example for a beta test such as this, the tester states the nonexistence of a screen text, navigational attribute or a program function although it actually exists. Exhibits 9-10 describe all known Type I and Type II errors made by the game's testers.

Exhibit 9
Tester Type I Errors

Error	Count
Did not note the column label for South America read “Mexico”.	18
Did not see the Earnings/Deficit in the Accounting window did not match the Net Income reported for that period.	18
Did not detect the Prior Period’s Retained Earnings/Deficit was not reported correctly.	18
No players noticed in the game’s Newspaper the current quarter’s results were not being reported but instead were showing their firm’s Year-To-Date Performance Index.	18
No firm noted the Help topic for “Market Demand” was labeled “Country Demand”. There are only continents or markets rather the countries in the game.	18
Did not notice Help mislabeled Market Area decisions as Country Market Decisions.	18
Did not note Help mislabeled their company’s financial center by country rather than by market.	18

Exhibit 10
Tester Type II Errors

Error	Count
Firm 6 states it would be good if the simulation allowed players to juggle windows. This feature exists under “Click here to open a new window” in the game’s tool bar. It is also explained in the Help topic “Screen View”.	1
Firm 1 states the Income Statement’s shipping expense is incorrect. The player is overlooking the shipping costs associated with importing the factory’s raw materials. This information could have been obtained by retrieving the entry’s Accounting Window by clicking on the account in the Income Statement.	1
Firm 1 claims the unit sales forecast for scooters in North America is too high when compared to the output generated by the game’s demand forecasting tool. The player is confusing total North American demand versus the demand for the firm’s specific set of scooters.	1

Based on this information those who filed their reports made 126 Type I errors and three Type II errors and therefore the hypothesis that no errors would be committed is rejected. The testers were very good at pointing out what they felt were grammatical errors although none of them discovered any of the simulation’s mathematical errors even though the reward structure for doing so was 20 times greater than that for detecting spelling and grammar errors.

DISCUSSION

Before discussing this test’s results, we should first analyze the degree the test’s design met the criteria for a good beta test because it is possible the beta test’s design or implementation was flawed. Exhibit 11 shows a scoring of the test’s design against the beta test ideal formulated for this study.

This exhibit’s results indicate the publisher created a valid testing situation except for the subjects involved. This means it almost completely conformed to what is necessary to bring about a good beta test. Therefore, it could be presumed any breakdown between what the ideal beta test should produce versus its reality, lies somewhere else.

A further analysis was conducted to determine what factors might have led to the testers failing to submit their game write-ups. This is an important analysis, as this was really what the beta test wanted to obtain. A multiple regression analysis on the submission of a report was conducted using as predictor variables, time spent with the game online, gender, company performance and class standing. The predictive value of this combination of potential predictors was nonsignificant after adjusting its r-square of 0.40 for the test‘s small sample size (Adjusted r-square = 0.22, F-statistic = 2.18). Accordingly, other factors are associated with report non-filing and they should be investigated and corrective measures taken if possible.

What is the cost of these errors? It would be good to be able to assign a cost so that the publisher could determine how many more beta tests should be performed to eliminate all possible costly errors. The true “costs” might be the user dropping the game thereby losing future royalties and earnings. It might be a negative standing in the marketplace or a reputation for releasing faulty software. It might be the extra cost or burden on the publisher’s Support group. On the other hand, software glitches and bugs, known as Schroedinbugs can lie dormant in software and never come

Exhibit 11
Ideal Beta-Test Design Components vs. Actual Conditions

Ideal	Actual	Conformity to the Ideal
Conducted by independent participants	The participants were paid players not under the direct employment of the game's publisher.	Perfect conformity for the game's testers.
Conducted by an objective administrator	1. The test's administrator was a consultant who specialized in conducting beta tests. This administrator also monitored turn-in conformance and e-mailed laggard participants and teams. No coaching was provided by this administrator. 2. The game's author acted as a team coach as a substitute for the normal role taken by an instructor using the game. The author provided active coaching for the game's first three rounds and coaching upon request for the test's final three rounds.	1. The consultant's responsibility was to enable a good test but was disinterested about its outcome. 2. The game's author wanted the test to be successful so that useful feedback could be obtained. 3. The game's author served as a proxy for the role expected of any instructor using the game. Verdict—High conformity.
Played by the application's target population	1. Testers were business school sophomores rather than the game's targeted freshmen. 2. Testers were naïve game players like the game's targeted population. 3. Most testers were Sophomores with the remainder being Juniors. 4. Testers were not randomly obtained from the university's student population within its business school. 5. The majority of the testers were members of a college-wide International Business Honors program	1. Testers were business school students that mirrored the game's target population. 2. Testers were naïve game players, which mirrored the game's target population. 3. Testers were not unschooled business school freshman, which does not mirror the game's target population. 4. More than 50.0% of the testers where enrolled in an honors program that required a GPA equal to or greater than 3.25 which is not the GPS expected of entry-level Freshmen. 5. Participants were interested in international business. This agrees with the game's orientation that is international in its scope. Verdict—Moderate conformity but non-conformity in a crucial areas.
Verification the application meets its intended purpose	Testers were asked whether they felt the game was appropriate for Freshmen	High conformity.
Specify test procedures and schedules	Testers were provided a welcoming letter, a Player's Guide that indicated how a company could make its decisions, a statement of the beta test's purpose, its schedule of events and company assignments.	High conformity.
Beta-testers receive remuneration	Testers were paid a staggered hourly rate and paid for every spelling and grammatical error	Not known whether the total and cumulative monetary rewards

	cited and a larger rate for every math error found.	provided high incentives for active participation. Testers were paid for every grammatical and spelling error found. An amount that was 20 times larger was awarded for each math or calculating error found. Verdict—Moderate conformity for the rates paid.
Determine and state the tester's role	Players were informed about what the test was to accomplish and their role was in bringing about those accomplishments.	High conformity.
Determine the simulation's expected values	The simulation's accounting and operations are known as well as the form those results should take.	This beta test was more a test of the game's new interface rather than its source program that was in its fourth generation. Verdict—High conformity.
Provide an effective and convenient reporting system for defects and suggestions	1. Players interfaced via the game's website as well as reporting results via e-mails. 2. All testers thoroughly familiar with online etiquette and procedures.	High conformity.

Exhibit 12
Potential Costs Associated with Releasing an Imperfect Game

Case	Scenario	Monetary Cost
A	1. Instructor asks for a "work around" for the problem 2. Instructor continues to use the game as long as a pizza party is provided to the class as an apology.	1. No cost for the work around. 2. Low cost for at $90.00 for the pizza party but future revenues undetermined.
B	1. Instructor demands a refund of all student game licenses 2. Instructor uses the game again.	1. Refund cost high in the current semester-- $540.00. 2. Undetermined revenues for future adoptions.
C	1. Instructor demands a refund of all student game licenses 2. Instructor never uses the game again.	1. Refund cost high in the current semester-- $540.00. 2. Sales revenue loss high in future semesters but loss undetermined.
D	1. Bug is fixed within three to five business days and patch installed. Game suspended for this one user while fix is created. 2. This bug and its delay cause students to begin to suspect the game experience's validity. 3. The instructor never uses the game again	1. Bug fix cost relatively high in the current run-- $240.00. 2. Negative effect on learning potential moderate but undetermined. 3. High cost in future semesters but total revenue loss undetermined.
E	1. Bug fixed overnight and patch installed. 2. Students suspect the game's validity as a learning method. 3. The instructor never uses the game again. 4. Instructor tells many colleagues the game "has problems".	1. Bug fix relatively low for the current run-- $120.00. 2. Negative effect on learning potential moderate but cost undetermined. 3. High cost in future semesters but total revenue loss undetermined. 4. Cost of instructor's opinion voiced to colleagues undetermined but conditioned by contacts and reputation in the field.

to the surface, or are never recognized by adopters until the simulation is used beyond its normal limits.

Given the imperfections found here, and those that probably are associated with any beta test, what criteria should be used by a publisher as to when to test more or go to market? No matter what could have been done with the study's software there are problems probably lurking in its software waiting to be discovered. Unfortunately, it is typically not economically feasible to continue testing until all defects are found and corrected.

One possible solution as to when to "close" the beta test and "release" the software product may be determined by the economic and finance literature's net present value calculation (NPV). Nevertheless, because the information from any beta test is imperfect, the NPV methodology cannot be applied, as it requires the "expected" costs and benefits to be identified and the assignment of associated expected costs to each possible outcome. Below is a list of possible scenarios and potential costs that may occur after a business simulation game is released to illustrate the problems of applying NPV solutions to the release dilemma. These examples are taken from the experiences known by this paper's authors. Given all these imponderables, the NPV rule cannot be accurately applied to any of the scenarios presented in Exhibit 12.

What else is available to the well-meaning game publisher? Other common approaches include payoff tables, decision trees or simulation models. The use of a payoff table requires a specification of the possible states of nature and the alternative actions that could be taken by the software publisher. A simple example would be the decision to release or not to release a software product. The states of nature could be the product is completely successful, the product has minor flaws and the product has major defects. The expected net benefits and probability of occurrence of each state of nature for each decision would then be assigned, and the resulting payoff in each of the cells in the payoff matrix determined. The decision with the highest expected net benefit would be selected but this would be a futile exercise and the decision maker does not know *a priori* the probabilities of the states of nature.

The decision tree method comes across the same specification problems but in different forms at different junctures. Again, the expected net benefits and the probabilities associated with each sequence of events must be assigned. The simulation approach is probably the most involved and requires the set of possible scenarios to be simulated and the outcomes calculated. A sophisticated simulation would allow the cash flows to be calculated for numerous alternatives but only if the cash flows could be accurately calculated and predicted.

Thus, a software go to market decision is basically a "best feeling" situation. The publisher should not knowingly distribute untested or poorly tested software, but even the wealthiest software firms attach no liability warranties on their products. This appears to be the case for the business game that was this study's focus. The firm's president, a marketing professor himself, had full knowledge of the vagaries of product market testing but wanted to do as much as possible to minimize, and hopefully eliminate, all user software-related problems.

CONCLUSION

It is well established that an effective Beta test is necessary before any software application is released to the market. An effective beta test will help a software developer meet the important market release conditions of targeted audience propriety, playability, model fidelity and algorithmic accuracy. Because of these conditions it is critical for software developers to fully comprehend the beta testing process.

To better understand the nature and challenges of implementing this process, a beta test was created and evaluated for a first-generation online business game by a major online game publisher. The example beta test presented in this paper was designed to meet the ideal beta test criteria comprised of qualifications of the participants, the requirements list of the application, the test procedures, reporting systems, and the defect analysis.

Although an effective beta test process was followed, problems arose with respect to the participation of the testers. Not all testers participated equally and sufficiently with respect to their involvement with the game and with providing adequate written feedback. This occurred despite the fact that the testers were provided financial incentives to do so, as recommended in the beta testing literature. The reason for this behavior was examined and could not be found to be related to the testers' performance on the game, gender, or class standing. Perhaps greater incentives or different incentives are needed to promote tester involvement, including greater involvement and encouragement by the test game administrators. This problem of uneven, within-team participation is not, however, unique to this study. Participation in group projects is rarely equal and a similar participation counting method for a similar game found equally-low participation rates (Wolfe & McCoy, 2008). A study by Fine (2002) recommends simple but non-monetary gifts such as T-shirts or coffee mugs whereas this study used a financial reward system. We recommend that it is important to understand the incentives, *a priori*, that would work for the specific testers in a beta study. More research is warranted on this issue.

Another issue is concerned with the quality of the tester's feedback. A review of the tester feedback in this study indicated a high degree of Type I and Type II errors. This problem was not found to be due to the qualifications of the testers in our study. These testers had the background and capabilities necessary to evaluate the business simulation game. This problem could be related to the apparent lack of motivation of several of testers as evidenced by their inadequate written feedback. Again, a revised incentive system might help here. This is a serious concern that may exist with many beta testing situations and clearly warrants further research.

The final stage of the beta testing cycle, the "closure" or "release condition" of the software application was found to be a contentious issue. In our case study, a well-designed beta test could not guarantee the release of error free software. Yet, it is generally not feasible to continue testing until all defects are found and corrected. To help make the "closure" decision, the economic paradigm of applying techniques such as NPV, Payoff Tables, Decision Trees Analysis, and even Simulation was reviewed. It is believed it would be futile to use these techniques given the imponderables associated with measuring the expected benefits, expected costs and the probabilities of the various outcomes. Further research is needed that address the application of economic tools of analysis for better quantifying release conditions. With respect to "closure", we recommend a final meeting take place with the stakeholders to address the results of the beta test, review lessons learned and discuss the advantages and disadvantages of releasing a product or continuing development and re-testing. The decision to release or re-test a product needs to be based on a careful balance between budget considerations, quality issues, risk assessments and time availability given the uncertainties associated with the entire process.

REFERENCES

Bach, J. (1999), Risk and Requirements- Based Testing, *Computer, 32*(6): 113-114.

Byers, C. and Cannon, H. M. (2007). The Programming Game: An Exploratory Collaboration Between Business Simulation and Instructional Design, *Developments in Business Simulation and Experiential Learning, 34: 259-265*.

Dowson, M. (1997). The Ariane 5 Software Failure. *Software Engineering Notes, 22*(2): 84.

Fine, M. (2002), *Beta Testing for Better Software*, Indianapolis: Wiley.

Harmesh, D. (2009) Beta Testing, Anyone? Ten Potent Strategies for Achieving Success, Articlesbase August 4, 2009. http://www.articlesbase.com/computers-articles/beta testing-anyone-ten-potent-strategies-for-achieving-success-1095782.html.

Kaner, C. Keynote address, Exploratory Testing. *Quality Assurance Institute Worldwide Annual Software Testing Conference*, Orlando, FL, November 2006.

Kaner, C., Bach, J. and Pettichord, B. (2003), *Lessons Learned in Software Testing: A Context-Driven Approach*. Indianapolis: Wiley.

Levenson, N.G. and Turner, C.S. (1993). An Investigation of the Therac-25 Accidents. *Computer*, 26(7): 18-41.

Pan, J. (1999), Software Testing (18-849b Dependable Embedded Systems), *Topics in Dependable Embedded Systems*, Electrical and Computer Engineering Department, Carnegie Mellon University, http://www.ece.cmu.edu/~koopman/des_s99/sw_testing/

Rogerson, S. (2002). The Chinook Helicopter Disaster. *IMIS Journal*, *12*(2): ETHIcol.

Shea, G. (2006), Better Beta: The more you know about beta testing, the more your company will gain from the experience. *Computer World*, *40*(5): 43-44.

Wolfe, J., and McCoy, R. (2008). Should business game players choose their teammates: A study with pedagogical implications. Paper presented, Association for Business Simulation and Experiential Learning, Charleston SC.

Yang, M.C.K. and Chao, A. (1995). Reliability-estimation and stopping-rules for software testing based on repeated appearances of bugs. *IEEE Transactions on Reliability*, *44*(2): 315-21.