

2011

# The Validity and effectiveness of a business game beta test

Steven Gold

Joseph Wolfe

Follow this and additional works at: <http://scholarworks.rit.edu/article>

---

## Recommended Citation

Gold, Steven and Joseph Wolfe. "The Validity and effectiveness of a business game beta test," *Simulation and Gaming* (2011). Web.

This Article is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Articles by an authorized administrator of RIT Scholar Works. For more information, please contact [ritscholarworks@rit.edu](mailto:ritscholarworks@rit.edu).

# Simulation & Gaming

<http://sag.sagepub.com/>

---

## The Validity and Effectiveness of a Business Game Beta Test

Steven C. Gold and Joseph Wolfe

*Simulation Gaming* published online 14 December 2011

DOI: 10.1177/1046878111431868

The online version of this article can be found at:

<http://sag.sagepub.com/content/early/2011/12/13/1046878111431868>

---

Published by:



<http://www.sagepublications.com>

On behalf of:

Association for Business Simulation & Experiential Learning

International Simulation & Gaming Association

Japan Association of Simulation & Gaming

North American Simulation and Gaming Association



Playful Methods. Serious Results

North American Simulation & Gaming Association

Society for Intercultural Education, Training, & Research

**Additional services and information for *Simulation & Gaming* can be found at:**

**Email Alerts:** <http://sag.sagepub.com/cgi/alerts>

**Subscriptions:** <http://sag.sagepub.com/subscriptions>

**Reprints:** <http://www.sagepub.com/journalsReprints.nav>

**Permissions:** <http://www.sagepub.com/journalsPermissions.nav>

>> [OnlineFirst Version of Record](#) - Dec 14, 2011

[What is This?](#)

---

# The Validity and Effectiveness of a Business Game Beta Test

Simulation & Gaming  
XX(X) 1–25  
© 2011 SAGE Publications  
Reprints and permission:  
sagepub.com/journalsPermissions.nav  
DOI: 10.1177/1046878111431868  
<http://sag.sagepub.com>



Steven C. Gold<sup>1</sup> and Joseph Wolfe<sup>2</sup>

## Abstract

New gaming software must undergo a series of tests before its general release. The objective of these tests is to ensure that 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 article first catalogs the design parameters associated with a good beta test. It then compares this ideal against the beta test created for a first-generation online business game released by a major online game publisher. It then examines the actual behaviors and results produced by the study's beta testers to determine the degree the publisher and authors could be confident that 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, and the likely reasons for this outcome are identified.

## Keywords

algorithmic accuracy, audience propriety, beta testing, beta-test procedure, business simulation development, business simulation testing, defect analysis, model fidelity, playability, release conditions, software development, software testing, test design

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 are annoyances, but they are minor and inconsequential compared with 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

---

<sup>1</sup>Rochester Institute of Technology, Rochester, NY, USA

<sup>2</sup>Experiential Adventures LLC, Bainbridge Island, WA, USA

## Corresponding Author:

Steven C. Gold, Saunders College of Business, Rochester Institute of Technology, 106 Lomb Memorial Drive, Rochester, NY 14623-5603  
Email: [stevengold@saunders.rit.edu](mailto:stevengold@saunders.rit.edu)

radium overdoses (Levenson & Turner, 1993). An onboard program error caused the destruction of the Ariane 5 prototype 1 min into its flight on June 4, 1996, at the cost of more than US\$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 (Rogerson, 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. The programming demands for online business games are even higher because they not only include source code for the game, but for the active interface as well. This interface must be attractive, be "open," and anticipate every false move made by its players. 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 both *Simulation & Gaming: An Interdisciplinary Journal* and the *Bernie Keys Library*, a research archive containing a complete word-searchable compilation of all past Association for Business Simulation and Experiential Learning (ABSEL) proceedings published in *Developments in Business Simulation and Experiential Learning*, reveals the mechanics of conducting a valid beta test are not discussed nor has a discussion taken place of what is revealed and not revealed by conducting a beta test. When mentioned at all, game authors either state their game is being or was beta tested, or that their beta tests were encouraging (Nissen, 1996; Prakash et al., 2009; Thorelli, 2001). Byers and Cannon (2007) discuss how a beta test fits in the game design and development process, but do not discuss details concerning the effective implementation of such a test.

The lack of research on beta-testing methodologies with respect to the development of business simulations and games is a noteworthy deficiency owing to the key role that it plays in this process. This article attempts to direct the attention of simulation developers to this critically important area by examining the creation and conduct of an actual beta test as well as serving to open a debate on the realities of conducting effective beta tests. To do that, this article reviews the nature of the software testing cycle with a special emphasis on the beta-test phase. It then outlines the qualities that should be present to ensure that a beta-test produces 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

The literature on the beta-testing process for software development is wide and deep (Beizer, 1990; Gelperin & Hetzel, 1988; Srinivasan & Gopalaswamy, 2006; Whittaker, 2000). 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 becomes 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 general availability (GA) or *gold code* for the gold standard expected of released software.

Pan (1999) identifies the key steps in a typical beta-testing process. These steps are the following and are discussed in detail below:

- Requirements Analysis
- Testing Procedures
- Reporting Systems
- Defect Analysis and Retesting
- Closure

### *Requirements Analysis*

A critical 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, two significant customers are 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):

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. (p. 114)

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 article, 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, and perhaps the most important component, is the selection of the participants or subjects. Kaner (2006) highlights the importance of selecting independent participants who are managed by objective test administrators. Equally important is that the participants (testers) be characteristic of the game's target population (users). To achieve objectivity in the test's administration, no incentives, direct or indirect, should be given to 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. A mistake would be to use "friends" of the authors. Even though they may provide a serious and responsible review of the simulation game, they would not meet the critical conditions of objectivity and being characteristic of the end user (customer). However, during alpha testing stage, this would be a useful source of feedback to the developers of the game.

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 as 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, predetermine the expected outcomes so that these outcomes can be compared with 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).

## **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 email 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 and timely, and minimizes the probability of not receiving relevant information.

## *Defect Analysis and Retesting*

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.

## *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 and Chao (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.

## *Summary of Key Beta-Test Design Process Components*

Based on a review of the recommended beta-testing process, Table 1 summarizes the key components required for a thorough and complete test that we have found in the literature. The requirements list is the first critical step and is needed to verify that the application meets its intended purpose, and to help determine the qualifications of the participants, including the testers, administrators, and the target population. 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 cancel the entire release.

**Table I.** Ideal Beta-Test Design Components

---

Requirements list	Verify the application meets its intended purpose
Qualifications of participants	Conducted by independent testers
	Conducted by an objective administrator
	Played by the application's target population
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

---

### *Setting Up the Beta Test: Process and Method*

The game being tested was *THE GLOBAL BUSINESS GAME: BUSINESS BASICS EDITION* (Wolfe, n.d.). It is a simplified version of *THE GLOBAL BUSINESS GAME: WORLD EDITION*. Its original source code has been used since its December 2000 release. 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 whether

- its inherited source code was error free,
- the author had written the right software for its intended audience, and
- the author had 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 subassembly kit imported from Asia
3. two continental markets operating under stable economic 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 beta test was set up to follow the best practices identified in the literature review and was not constrained by time or resources. The beta test consisted of the six steps identified in the literature.

*Step 1: Specifying the requirements list.* The requirements list details the key objectives and desired outcomes of the beta test. The primary objectives of this beta test were to find out whether the game was viewed, by students, as suitable for an Introduction to Business class and how easy it was for them to play and enjoy the game while engaged in its subtle learning process. Table 2A 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?" It is important to note that the reference to the game's software includes the game's interface and the code that was written to enhance the user interface. The requirement's list also dealt with the software's stability, performance, and market/customer reach given its intended audience. Given these objectives, the following questions were posed to the beta testers.

Table 3 summarizes the ideal versus actual characteristics of the requirements list and the conformity to the ideal. This step in the procedure was considered to have high conformity to the ideal beta test.

*Step 2: Selecting qualified participants (players and administrators).* The beta test involves the selection of participants (players) and the game administrator, and as much as possible duplicating the environment under which the game will be played, that is, the classroom and its target player audience. In all, 18 business school sophomores at a large southern university served as the study's beta testers. Eighteen students is a reasonable number for a test, but is smaller than the target market for an Introduction to Business class. The "ideal" beta test should be with a class size as close as possible to the target market environment. This group-test size was selected, however, because it was comparable with those previously used by the publisher of similar games in the firm's portfolio of simulations.

To be a part of the test, student participants had to meet two important criteria: (a) independent of the publisher of the game and (b) characteristic of the game's target population (users). The administrators of the game should be knowledgeable of the mechanics of using the game and the beta-testing procedure. To simulate a classroom environment, it is recommended that an administrator would also play the role of the instructor.

Table 3 summarizes the ideal versus actual characteristics of the participants and administrators, and the conformity to the ideal. The selection of the participants and administrators are described in the table and shows a high conformity with the ideal beta-test conditions. The participants consisted of business school students and

**Table 2A.** Requirements List—Respondent's Questions

Number	Question
1	How suitable is the game for an Introduction 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?

**Table 2B.** Requirements List and Verification of Objectives—Ideal Versus Actual

Ideal	Actual	Conformity to the ideal
Develop a list of detailed requirements of the application and how it will meet its intended purpose	Testers were asked a set of questions on whether they felt the game was appropriate for an Introduction to Business course and the ease of use	High conformity

inexperienced game players which mirrors the target market. Experienced beta testers or game players were specifically not chosen even though they may be more adept at finding defects, but would not represent the target market. However, the students used in the study were not freshman, but were relatively naïve regarding the entire range of business and accounting concepts associated with the game. This does represent a variance from the ideal selection of beta-test participants because the target market is Introduction to Business and typically has a large percentage of freshman students.

*Step 3: Establishing procedures.* Before play began, the testers were told the expectations for their role in the beta test and received a copy of the game's 10-page *Player's Guide*, an introduction email from the game's administrator with instructions on how to access the game at its web address, the study's questionnaire with instructions for its completion, their license number and game password, and the name and email address of their company's partner.

The procedure for the beta test involved the participants in the following ways:

- dedicating four consecutive weeks to play six decision rounds of a relatively simple online business game
- being one member of a two-member company
- submitting team decisions twice a week, by 8:00 p.m. every Monday and Thursday in a 3-week period
- providing endgame feedback via a structured questionnaire
- accepting and receiving the beta test's remuneration terms

**Table 3.** Participants in Beta Test—Ideal Versus Actual

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.
Played by the application's target population	<ol style="list-style-type: none"> <li>1. Testers were business school sophomores rather than the game's targeted freshmen.</li> <li>2. Testers were naive game players like the game's targeted population.</li> <li>3. Most testers were sophomores with a small number being juniors.</li> <li>4. Testers were not randomly obtained from the university's student population within its business school.</li> <li>5. The majority of the testers were members of a collegewide International Business Honors program.</li> </ol>	<ol style="list-style-type: none"> <li>1. Testers were business school students that mirrored the game's target population.</li> <li>2. Testers were naive game players that mirrored the game's target population.</li> <li>3. More than 50.0% of the testers were enrolled in an honors program that required a grade point average (GPA) equal to or greater than 3.25 which is not the GPAs expected of entry-level Freshmen.</li> <li>4. Participants were interested in international business. This agrees with the game's orientation that is international in its scope.</li> <li>5. Participants were not academically naive as none were incoming freshmen.</li> </ol> <p>Verdict—Moderate conformity, but nonconformity in a crucial area (Item 5).</p>
Conducted by knowledgeable administrators	<ol style="list-style-type: none"> <li>1. The test's administrator was a consultant who specialized in conducting beta tests. This administrator also monitored turn-in conformance and emailed laggard participants and teams. No coaching was provided by this administrator.</li> <li>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 on request for the test's final three rounds.</li> </ol>	<ol style="list-style-type: none"> <li>1. The consultant's responsibility was to enable a good test, but was appropriately disinterested about its outcome.</li> <li>2. The game's author wanted the test to be successful so that useful feedback could be obtained.</li> <li>3. The game's author served as a proxy for the role expected of any instructor using the game.</li> </ol> <p>Verdict—High conformity.</p>

The length of time of 4 weeks was selected because Introduction to Business classes would probably only devote a month of a semester course to the game, perhaps the closing month. In this sense, the beta test meets the condition of modeling the expected classroom scenario. The team size of two is within the range of the standard team size of two to four students, but three or four students per team may have been a more representative choice. Given the smaller number of students, it was decided that having smaller teams would allow for more competition, which would better reflect the typical class game environment. Submitting two decisions per week is representative of the suggested time frame for the game.

Student testers were given monetary incentives to play the game seriously and meet their responsibilities, as recommended for an effective beta test. Yet, the amount of the incentive is not well described in the literature. Because of this, determining the incentive scheme is challenging and appears to be a judgment call. In this beta test, participants received compensation at .25 cents for every spelling and grammar error cited and US\$5.00 for every math error reported. The testers were also given a staggered hourly budget at US\$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 US\$7.25 per hour with part-time retail sales clerks earning an average wage of \$7.82 in the test's previous year. The authors are aware that some developers give extra credit points to students in a classroom setting as an incentive to participate, but we have not found research on this approach and its effectiveness.

Table 4 summarizes the ideal procedural components in a beta test compared with the actual components. The qualitative assessment showed high conformity in the first two criteria, but only moderate conformity with respect to the remunerations given to the participants. The assessment of moderate is owing to the lack of adequate guidelines in the literature and the fact that the incentives were close to the state's minimum wage, which is not a significant incentive.

*Step 4. Providing an effective reporting system.* In the beta test, participants could communicate and record their findings in two ways. The first was directly through the game's website interface. The second approach was through email. The testers were informed with respect to the procedures to use the website interface and email, the importance of fully and quickly reporting all defects or issues, and communication etiquette. Both methods of communication were convenient and allowed for real-time reporting of problems. As summarized in Table 5, we assessed the reporting system as having high conformity with the ideal. The students raised no questions or issues with respect to communicating their findings. Almost all testers filed their required reports, although some were slower at doing it and some were less detailed than others.

*Step 5. Evaluating results—Defect analysis.* Evaluating results first requires that the game's expected values be known. The expected values in the beta test were well known as the game being tested was *THE GLOBAL BUSINESS GAME: BUSINESS BASICS EDITION*, which is a simplified version of *THE GLOBAL BUSINESS GAME: WORLD EDITION*, now in its fourth edition. The source code used in the beta test was

**Table 4.** Procedure in Beta Test—Ideal Versus Actual

Ideal	Actual	Conformity to the ideal
Determine and state the tester's role	Players were informed about what the beta test was to accomplish and their role was in bringing about those accomplishments.	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.
Model test procedures to reflect the target market experience.	The length of time to play the game was 1 month, with two students per team. Two decisions were required per week.	This scenario meets the typical class environment. Verdict—High conformity.
Beta testers receive remuneration	Testers were paid a staggered hourly rate and paid for every spelling and grammatical error cited and a larger rate for every math error found.	Not known whether the total and cumulative monetary rewards 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.

**Table 5.** Reporting System—Ideal Versus Actual

Ideal	Actual	Conformity to the ideal
Provide an effective and convenient reporting system for defects and suggestions	<ol style="list-style-type: none"> <li>1. Players interfaced via the game's website as well as reporting results via emails.</li> <li>2. All testers thoroughly familiar with online etiquette and procedures.</li> </ol>	High conformity.

the same as that used by the game in its earliest iterations. Next, the expected values can be compared with the actual outcomes from the study. It is important that the defects reported by the testers are not just accepted when evaluating the success of the beta test. It is required, in the ideal beta test, to carefully evaluate the reported defects for both Type I and Type II errors to confirm their accuracy (Table 6).

**Table 6.** Defect Analysis—Ideal Versus Actual

Ideal	Actual	Conformity to the ideal
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.
Compare expected with actual values and evaluate for Type I and Type II errors	All reported defects were compared with expected values and evaluated for Type I and Type II errors as illustrated in Tables 7 and 8.	Verdict—High conformity.

In a statistical test, a Type I error is one where it is stated that 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 that the condition is False when it is actually True.

The evaluation of Type I and II errors were classified and the results are detailed in Tables 7 and 8. It was found that the testers (student participants) who filed their reports made 126 Type I errors and 3 Type II errors. 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 much greater than that for detecting spelling and grammar errors. We were surprised and concerned to find the significant number of Type I errors. This clearly highlights the extreme importance of checking for these types of errors and not just accepting the defects reported by the testers. Suggestions on how to minimize these types of errors are discussed later in the article.

*Step 6. Determining closure.* Given the imperfections found here with respect to Type I and II errors, and those that probably are associated with any beta test, what criteria should be used by a publisher to determine when to test more or go to market? No matter how thorough the beta test, problems probably lurk in the game's 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 accurately 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 article's authors. Given all these imponderables, the NPV

**Table 7.** 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." Only continents or markets are there rather than 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
• Did not notice that the Player's Guide displayed six continents even though only two are available.	18

**Table 8.** 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 with 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

rule was found to be difficult to utilize confidently to any of the scenarios presented in Table 9.

What else is available to the well-meaning game publisher? Other common approaches include payoff tables, decision trees, and 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

**Table 9.** Potential Costs Associated With Releasing an Imperfect Game

Case	Scenario	Monetary Cost
A	<ol style="list-style-type: none"> <li>Instructor asks for a “work around” for the problem</li> <li>Instructor continues to use the game as long as a pizza party is provided to the class as an apology</li> </ol>	<ol style="list-style-type: none"> <li>No cost for the work around</li> <li>Low cost at US\$90.00 for the pizza party, but future revenues undetermined</li> </ol>
B	<ol style="list-style-type: none"> <li>Instructor demands a refund of all student game licenses</li> <li>Instructor uses the game again</li> </ol>	<ol style="list-style-type: none"> <li>Refund cost high in the current semester—US\$540.00</li> <li>Undetermined revenues for future adoptions</li> </ol>
C	<ol style="list-style-type: none"> <li>Instructor demands a refund of all student game licenses</li> <li>Instructor never uses the game again</li> </ol>	<ol style="list-style-type: none"> <li>Refund cost high in the current semester—US\$540.00</li> <li>Sales revenue loss high in future semesters, but loss undetermined</li> </ol>
D	<ol style="list-style-type: none"> <li>Bug is fixed within three to five business days and patch installed. Game suspended for this one user while fix is created.</li> <li>This bug and its delay cause students to begin to suspect the game experience’s validity</li> <li>The instructor never uses the game again</li> </ol>	<ol style="list-style-type: none"> <li>Bug fix cost relatively high in the current run—US\$240.00]</li> <li>Negative effect on learning potential moderate, but undetermined</li> <li>High cost in future semesters, but total revenue loss undetermined</li> </ol>
E	<ol style="list-style-type: none"> <li>Bug fixed overnight and patch installed</li> <li>Students suspect the game’s validity as a learning method</li> <li>The instructor never uses the game again</li> <li>Instructor tells many colleagues the game “has problems”</li> </ol>	<ol style="list-style-type: none"> <li>Bug fix relatively low for the current run—US\$120.00</li> <li>Negative effect on learning potential moderate, but cost undetermined</li> <li>High cost in future semesters, but total revenue loss undetermined</li> <li>Cost of instructor’s opinion voiced to colleagues undetermined, but conditioned by contacts and reputation in the field</li> </ol>

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 with some supporting quantitative analysis. As we see it, it’s a judgment call on the part of the developer.

## The Quality of the Beta Test

Given high conformity of the beta-testing process to the ideal, we expected high reliability and quality with respect to the “responsible” participation of the testers (student players). The following hypotheses were tested to determine the reliability and quality of the beta test’s results, and to try to uncover the reasons for the high level of Type I errors by the participants (testers).

### Hypotheses

*Hypothesis 1:* All testers will actively participate in playing the game.

*Hypothesis 2:* All testers will be disciplined in their approach to the game.

*Hypothesis 3:* All testers will provide full and complete answers to the study’s feedback questions.

*Hypothesis 4:* A high correlation will be found between tester participation and the amount of feedback given.

*Hypothesis 5:* A high correlation will be found between participation and the number of hours billed by the testers.

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 emails 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.

1. All testers will actively participate in playing the game.

This study’s first hypothesis stated that all the game’s testers would actively participate in playing the game they were evaluating. Figure 1 presents a graph of the range of screen time minutes spent by decision 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. Figure 2 further indicates the amount of total within-team participation for all periods.

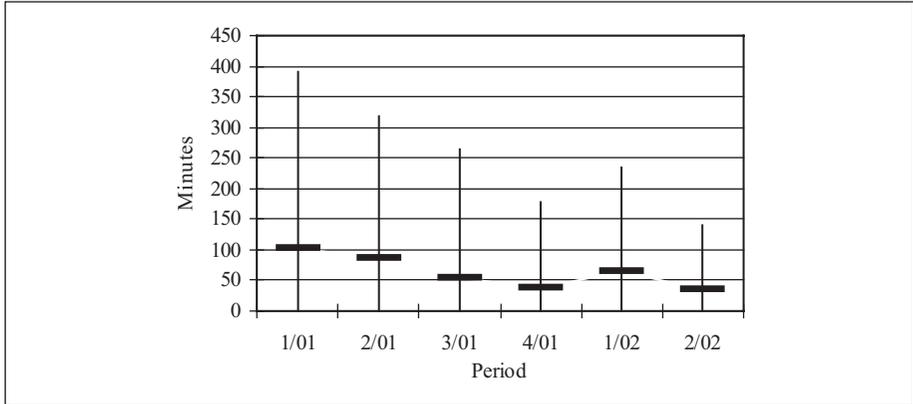


Figure 1. Screen time minutes by period

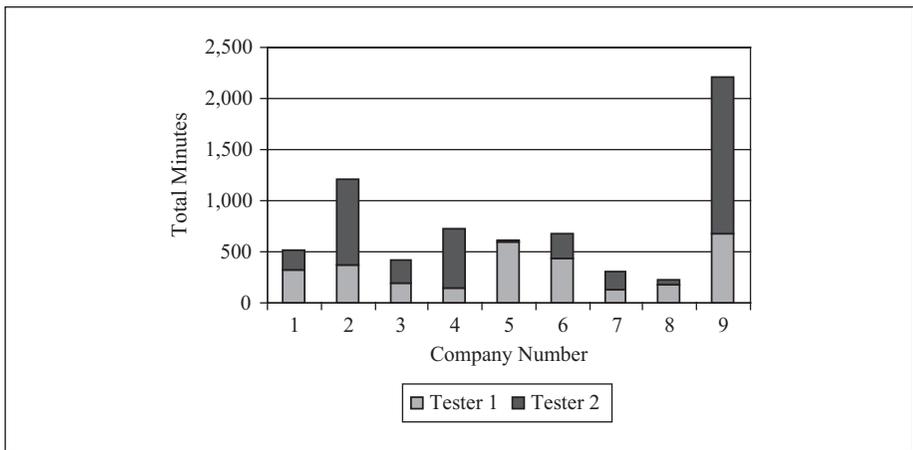


Figure 2. Total and individual screen time by team

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 min on that round’s decisions.

**Table 10.** Poor Discipline Incidents

Period	Incident
Pregame	A player from Firm 1 emails the Game Coach that he is not playing the game seriously.
Pregame	A player from Firm 4 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 2½ days late thereby holding up the entire game.
5	One player on Firm 9 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 2¾ days late.
5	Firm 9 submits its decision set 3 days late.
6	Firm 1 submits its decision set 1½ days late.

## 2. All testers will be disciplined in their approach to the game.

The study's second hypothesis stated that the testers would be disciplined or systematic in their conduct within the test's requirements. All email messages associated with the game were retained. Table 10 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 (a) the test's pacing had been announced in advance and (b) 3 to 4 days occurred 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 that the testers would give full and complete responses to the study's questionnaire. This hypothesis was tested by two methods. Table 11

**Table 11.** Responses by Question

Question	Count	%
How suitable is the game for an Introduction 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

**Table 12.** 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

shows the number of times each question was answered based on the 11 testers who responded and not on the 18 testers that *should* have responded.

3. All testers will provide full and complete answers to the study's feedback questions.

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. Table 12 indicates how complete each responding tester's answers were given the questionnaire asked six questions.

Taken in total, this hypothesis regarding complete responses must be rejected. Of the 18 testers, 9 either did not answer any question or did not respond to the

questionnaire. Of those who *did* respond, only 3 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 businesslike 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 rewordings, and 35 links to the sources of errant texts.

4. Tester participation will correlate highly with the amount of feedback given.

The fourth hypothesis looked to see whether a high ratio existed between the amount of time the testers put into playing the game and the amount of information they provided. It was assumed that 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 to the game, and the number of words found in their reports was low, but statistically significant with an  $r^2$  of .182 and a  $p$  value of .04 in a one-tail test. This weak correlation means that 81.8% of the variation in the size of their reports was associated with other factors.

5. Participation will correlate highly with the number of hours billed by the testers.

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. A positive, but weak relationship has already been determined between the amount of time the testers spent playing the game and the size of their reports. Table 13 shows the actual hourly rate the testers were paid based on the amount of time they spent online with the game and the pay per report word. The game pay rate per hour ranged from US\$0.00 to US\$20.31 with an average hourly rate of US\$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 US\$0.04 to US\$0.73 per word or an average of US\$0.27 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  $r^2$  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.

**Table 13.** Pay Per Online Game Time and Report Size

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

Note. N/R = no report.

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 actually the beta test's objective. 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  $r^2$  of .40 for the test's small sample size (adjusted  $r^2 = .22$ ,  $F$  statistic = 2.18). Accordingly, other factors are associated with the nonfiling of reports and they should be investigated and corrective measures taken.

## Discussion

An effective beta test is necessary before any software application is released to the market. A publisher should not knowingly distribute untested or poorly tested software. An effective beta test should help a software developer meet the important market release conditions of 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 ideal, the beta test used by a major game publisher was examined for a first-generation online business game. The example beta test presented in this article was designed to meet the ideal beta-test criteria comprising qualifications of the participants, the requirements list of the application, the test procedures, reporting systems, and the defect analysis.

Many of the indicators of an effective beta-test process were followed. However, problems arose with respect to the nature and participation levels of the testers used for the test. Not all testers contributed equally regarding their involvement with the game or with providing adequate written feedback. This occurred despite the fact that the testers were provided financial incentives to do so. Possible reasons for this deficient behavior were examined. None could be found that related to the testers' game performance, 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. A similar participation counting method for a similar game found equally low participation rates (Wolfe & McCoy, 2008). Other, nonmonetary incentives might have been effective. A study by Fine (2002) recommends simple gifts such as T-shirts or coffee mugs whereas this beta test used a financial reward system. An examination of a priori tester rewards and incentives important to tester candidates is warranted and should be subjected to their objective usefulness toward obtaining more-active tester involvement.

Another issue concerns the quality of the tester's feedback. A review of their feedback indicated a high degree of Type I and Type II errors. This problem was not due to the qualifications of the testers in our study. These testers had the background and capabilities necessary to evaluate the business simulation game as they had been exposed to its knowledge domain through prior coursework. One possible explanation could be that the data gathering instruments were not well utilized by the participants (testers). This, however, does not seem to be the likely cause based on their use of the online data gathering tools they had been exposed to in their training session. It is believed that the test's reporting problem is most likely related to the lack of motivation by several of the testers. This was demonstrated 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, which entails closure or release conditions, was also a contentious issue. In a practical sense, it is impossible to continue testing until all the new software's defects are found and corrected. No matter how much testing is done, we will always find errors at the extreme or in extreme cases. Thus, a conscientious publisher needs some type of standard or decision rule regarding when such testing should end. 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. Unfortunately, it has been found that these techniques are difficult to apply accurately given the imponderables associated with measuring the expected benefits, expected costs, and the probabilities of the various outcomes. It would be good to be able to assign costs or benefits so that the publisher could determine how many more beta tests should be performed to eliminate all possible costly errors. The true, long-term costs to the publisher of releasing bad software might be the user dropping the game thereby losing future royalties and earnings. Another cost could be the publisher earning a negative reputation for releasing faulty software. Another cost might be the extra cost or burden on the publisher's support group to help users deal with glitches and negative player reactions. However, software glitches and bugs, known as Schroedinbugs, can lie dormant in software and never come to the surface, or are never recognized by adopters until the simulation is used beyond its normal limits.

## Conclusion

This study examined the beta-testing program used by a major game developer and publisher. A number of problems arose that compromised the publisher's ideal method. Because problems will always be associated with even a perfect beta-test design, our study shows that further research is needed on the implementation process, especially in the area of effective tester performance. It is further suggested this research investigate the use of economic tools to determine the practical limits of conducting beta tests of analysis for better quantifying release conditions. With respect to closure, we recommend a final meeting take place with the stakeholders to address the beta test's results; review lessons learned, and discuss the advantages and disadvantages of releasing the product at that time. The decision to release or retest 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.

Simulation development may benefit from future research in the area of beta testing and release conditions. It is hoped that this article will serve as a catalyst for future research on beta testing owing to the paucity of research on this topic and its importance with respect to the effective development of business games.

## Declaration of Conflicting Interests

The authors declared the following potential conflicts of interest with respect to the research, authorship, and/or publication of this article: This article's second writer was the author of the game tested. Neither authors of this article had or have a personal or financial interest in the firm that conducted the beta test. The beta-testing firm had a contractual obligation to the game's publisher and neither of this article's two authors.

## Funding

The authors received no financial support for the research and/or authorship of this article.

## References

- Bach, J. (1999). Risk and requirements-based testing. *Computer*, 32(6), 113-114.
- Beizer, B. (1990). *Software testing techniques*. Boston MA: International Thomson Computer Press.
- Byers, C., & 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, IN: Wiley.
- Gelperin, D., & Hetzel, B. (1988). The growth of software testing. *Communications of the ACM*, 31, 687-695.
- THE GLOBAL BUSINESS GAME: BUSINESS BASICS EDITION. (n.d.). [Developed by J. Wolfe.]. Innovative Learning Solutions. Retrieved from <http://web3.onlinebg.com>
- Kaner, C. (2006, November). *Keynote address—Exploratory testing*. Quality Assurance Institute Worldwide Annual Software Testing Conference, Orlando, FL.
- Levenson, N. G., & Turner, C. S. (1993). An investigation of the Therac-25 accidents. *Computer*, 26(7), 18-41.
- Nissen, M. E. (1996). Designing qualitative simulation systems for business. *Simulation & Gaming*, 27, 462-483.
- Pan, J. (1999). *Software testing (18–849b Dependable Embedded Systems)*. Topics in Dependable Embedded Systems, Electrical and Computer Engineering Department, Carnegie Mellon University. Retrieved from [http://www.ece.cmu.edu/~koopman/des\\_s99/sw\\_testing/](http://www.ece.cmu.edu/~koopman/des_s99/sw_testing/)
- Prakash, E., Brindle, G., Jones, K., Zhou, S., Chaudhari, N. S., & Wong, K. (2009). Advances in games technology: Software, models, and intelligence. *Simulation & Gaming*, 40, 752-801.
- Rogerson, S. (2002). The Chinook helicopter disaster. *IMIS Journal*, 12(2), ETHicol. Retrieved from <http://www.ccsr.cse.dmu.ac.uk/resources/general/ethicol/Ecv12no2.html>
- 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.
- Srinivasan, D., & Gopalaswamy, R. (2006). *Software testing: Principles and practices*. New Delhi, India: Pearson Education.
- Thorelli, H. B. (2001). Ecology of international business simulation games. *Simulation & Gaming*, 32, 492-506.
- Whittaker, J. A. (2000). What is software testing? And why is it so hard? *IEEE Software*, 17(1), 70-79.
- Wolfe, J., & McCoy, R. (2008). Should business game players choose their teammates: A study with pedagogical implications. *Developments in Business Simulation and Experiential Learning*, 35, 318-328.
- Yang, M. C. K., & Chao, A. (1995). Reliability-estimation and stopping-rules for software testing based on repeated appearances of bugs. *IEEE Transactions on Reliability*, 44, 315-321.

**Bios**

**Steven C. Gold** is a professor of economics in the Saunders College of Business at the Rochester Institute of Technology. He is a fellow and past president of ABSEL, and has authored four computerized simulation games with such notable publishers as McGraw-Hill and Macmillan. His most recent simulation is designed to teach economics, called Beat the Market, and published with Gold Simulations.

Contact: [stevengold@saunders.rit.edu](mailto:stevengold@saunders.rit.edu)

**Joseph Wolfe**, PhD, Stern School of Business, New York University. He is a past ABSEL president and is an ABSEL fellow. He is the author of more than 45 case studies and a family of online business games.

Contact: [Jwolfe8125@aol.com](mailto:Jwolfe8125@aol.com)