



APRIL 25-30, 2010
ORLANDO, FLORIDA
ROSEN SHINGLE CREEK

T17

Track

4/29/2010 3:00 PM

"Keys to a Successful Beta Testing Program"

Presented by:

**Rob Swoboda
HomeShore Solutions**

Brought to you by:



330 Corporate Way, Suite 300, Orange Park, FL 32073
888-268-8770 · 904-278-0524 · sqeinfo@sqe.com · www.sqe.com

Rob Swoboda

Rob Swoboda has more than ten years of quality management and quality assurance experience. His diverse background includes work in the education, insurance, financial, banking, shipping, and biochemical industries. Rob has expertise in all levels of testing from unit to beta testing, testing methodologies, product and project management, and configuration management. Rob created and managed functional testing, systems integration, user acceptance, and beta teams whose effectiveness has resulted in a reduction of product delivery time and implementation issues by 50% to 66%.

I'm Speaking at
STAR EAST
APRIL 25-30, 2010

STAR EAST SOFTWARE TESTING ANALYSIS & REVIEW
2010

APRIL 25-30, 2010
ORLANDO, FLORIDA
ROSEN SHINGLE CREEK

I'm Speaking at
STAR EAST
APRIL 25-30, 2010

Keys to a Successful Beta Testing Program



Beta testing, Beta Process, and Beta Best Practices.

1

Introduction

STAR EAST SOFTWARE TESTING ANALYSIS & REVIEW
2010

APRIL 25-30, 2010
ORLANDO, FLORIDA
ROSEN SHINGLE CREEK

- Purpose
- Process
- Value
- Time

2

I'm Speaking at
STAR EAST
APRIL 25-30, 2010

STAR EAST SOFTWARE TESTING ANALYSIS & REVIEW
2010

APRIL 25-30, 2010
ORLANDO, FLORIDA
ROSEN SHINGLE CREEK

I'm Speaking at
STAR EAST
APRIL 25-30, 2010

Module 1 Beta Testing

3

Module 1 -
Beta Testing
Objective

STAR EAST SOFTWARE TESTING ANALYSIS & REVIEW
2010

APRIL 25-30, 2010
ORLANDO, FLORIDA
ROSEN SHINGLE CREEK

Provide an understanding of what Beta testing is

- Purpose/Value
- Types
- Structure and Process
- Beta teams, stakeholders and their needs

4

- Beta is a sub phase of the Software Development Life Cycle's Implementation phase
- Definition of Beta testing
- What is the purpose/value of Beta testing?
- Types of Beta testing

When preparing for a product's Beta phase address the following items:

- Beta Plan Document
 - Define scope and goals
- Beta Plan Meeting (Review of)
 - Set expectations for all stakeholders
- Beta Matrix
 - Structure the timetable and metrics
- Determine internal and external participants
- Install Beta release at Beta sites

Wait, there's more...

When preparing for a product's Beta phase address the following items (Continued):

- Set reporting/meeting schedules
- Identify onsite Beta partners
- Provide training if needed
- Create data collection and associated Beta documentation
- Compile data into final Beta Report
- Hold a post-release lessons learned meeting.

The following are some suggestions to ensure a successful and productive Beta:

- Well-defined, clear lines of communication
- Control of communications and perception of the Beta
- Assure customers that we have processes in place to deal with issues.
- Set clear, achievable goals for your Beta.
- Set length of Beta.
- Collect meaningful data.
- Ensure no regression issues were introduced.
- Keep the number of Beta sites manageable.
- Schedule regular reporting intervals.



- **Applicants for the Beta:** Ask qualified Beta candidates to apply for the Beta testing program
 - **Beta cycles:** Plan to release new builds to Beta participants once every two weeks to address identified issues
 - **Reset the Clock:** If you add a new feature or major fix to existing functionality during the Beta process, the clock goes back to the beginning of the Beta schedule
 - **Incentives:** are a great way of showing appreciation for your Beta partners but must be earned.
- ✓ *The Beta contract or agreement should also specify our company's obligation to support the Beta participant.*

9



- The number of required testers may vary.
- Some estimates report that only 20% of Beta users send back useful Beta information.
- Most Beta testers will try the program when they first get it, and then lose interest. **Keep your Beta testers motivated!**
- Most Information Technology personnel focus primarily on the technical aspects of the Beta release.
- At all times, the Beta team members and stakeholders must present the Beta product in the best possible light.
- The Beta director: Distribute pertinent information to all parties.

10



Choose effective Beta sites and Beta partners.
Key considerations in Beta participant selection
include the following:

- **Value:** What test items, unique configuration or conditions does the Beta site/user deliver?
- **Commitment:** Ask qualified Beta candidates to apply for the Beta testing program. This method gauges commitment.
- **Ability:** Assess the Beta participants' ability.
- **Attitude/Motivation:** A Beta participant may be committed to the process but still be motivated by factors that are not conducive to a fair evaluation of the product.

11



Nominations for the Beta release are submitted by the key product stakeholders.

The following is one example of a nomination process :

- Local support Engineers are given Beta Candidate acceptance criteria.
- They then tell their Account Executives to contact their District Managers about these possible qualified candidates.
- The Account Executives then builds a case for possible candidates.
- Nominated candidates are presented to the Core Team members representing the product.

12



- Nominees undergo an internal review.
- Meetings with the accepted nominees are held.
- A contract with the customer is signed.
- The logistical planning starts and resources are assigned to the implementation.



Module 2 Beta Documentation Set



The Beta Phase should have a specific Documentation and Artifact set. The extent and formality of the Beta Documentation Set will depend on the organization, product, and scope of the Beta. At a minimum every Beta should contain the following:

- ✓ Beta Test Plan
- ✓ Beta Matrix or Metrics tracking artifact
- ✓ Beta Report

The Beta Documentation and Artifact set are the tools used to **manage the beta process** , to set goals and expectations, to collect/distribute data/information, to gauge the effectiveness of the Beta and the accuracy of the Beta objectives (success factor) .



- The example Beta Documentation Set is more extensive but also scalable, depending on the scope of the Beta release it is being applied to.
- The full example Beta Documentation Set contains artifact for these four categories:
 - Internal artifacts
 - Internal emails
 - External or Customer facing artifacts
 - External or Customer facing email



We will now examine each of these artifacts:

- Link to Beta Templates
<http://teams.inside.XXX.com/tech/pet/etqa/Beta%20Template%20set/Forms/AllItems.aspx>



Internal artifacts

- Beta Test Plan.doc (A customer-facing Beta Guide may with other artifacts accompany the Beta Test Plan)
- Beta Matrix.xls
- Beta site candidate criteria form (may be part of the Beta Test Plan)
- Beta-Contacting Support Internal.doc
- Beta site ID.xls (Maybe included in the Beta Matrix.xls)
- Beta Roles and Responsibilities.doc
- Support Instructions.doc
- Beta Test report

The Beta Test Plan

- Introduction
- Beta Test Strategy and Methods
- Beta Scenarios
- Assumptions
- Beta Process
- Entry, Suspension, and Exit Criteria
- Resources
- Milestones and Test Deliverables
- Beta Test Items
- Release Documentation
- Variances
- Risk
- Summary

19

Beta Matrix.xls

- Beta Sites
- Sub Sites
- Site support Teams
- Builds
- Beta Wrap-up
- Actual Site Usage
- Incentive Summary
- Beta Issues Reported
- Beta Feedback
- Summary of Activities

20



- Beta site candidate criteria form
- Beta-Contacting Support Internal .doc
- Beta site ID .xls (Maybe included in the Beta Matrix .xls)
- Beta Roles and Responsibilities .doc (expanded and more detailed than Beta test plan section)
- Support FTP Instructions .doc

21



Beta Test Report

- 1) Introduction
- 2) Beta Summary
 - Beta Coverage Items and New Features/Fixes (NF/F)
- 3) Variances
- 4) Beta Timetable
- 5) Comprehensive Assessment
- 6) Summary of Activities
- 7) Risk and Open Issues
- 8) Test Report Document Signoff Form

22



- Internal emails
 - Initial Beta site confirmation email
 - Build announcement emails
 - Beta sites incentive emails
 - Scheduled status updates/progress reports emails
 - Internal Stakeholder survey emails

23



External or Customer facing artifacts

- Beta Site Acceptance letter .doc
- Beta Site Agreement / Contract .doc
- Beta Site Guidelines .doc
- Special data collection instructions (Report PDF – Win/Mac) .doc
- Beta Wrap-up Letter .doc

24



External or Customer facing email

- Initial Beta site contact email
- Build announcement emails
- Beta sites incentive emails
- Scheduled status updates/progress report emails
- Beta feedback form email

25



Module 3 Beta Testing Results

Conclusion

26



How do you measure the success or failure of your Beta experience?

- Did the Beta successfully satisfy all of the items within the scope of work?
- Can you claim success if the Beta has not successfully satisfied all of the items?
- In either of these cases, the product may not go to market.
- Note other factors that indicate success.
- Finally, from a product standpoint was it a successful Beta Phase?

27



- Success or failure of a Beta must be determined by the accuracy of the Beta objectives.
- A failed Beta can be detrimental to both your company and your company's clients.
 - For example, you may have met all of your objectives but failed to communicate effectively with your Beta participants and the Beta team.
 - In this case Product Management would deploy a product that the clients can't use and won't buy.
 - In the worst case, you may deploy a product that is extremely hard to support or that will fail and destroy the user's data.

28

Wrap-up /
Q&A



Questions?

StarEast Beta Test Methodology

Version 1.0

by

Rob L. Swoboda

February 2010

Contents

1.	Abstract.....	3
2.	Systems Development Life Cycle.....	3
3.	Beta Testing.....	4
3.1.	Beta Test Definitions.....	4
3.2.	Beta Best Practices	5
3.3.	Beta Structure	7
4.	Was the Beta Successful?.....	7
5.	Conclusion	8

Revision History

Name	Date	Reason for Changes	Version
Rob Swoboda	07/15/08	Initial Draft	0.1 draft 1
Rob Swoboda	07/20/08	Second Draft	0.2 draft 2
Rob Swoboda	07/20/08	Incorporate feed back Draft	0.3 draft 3
Rob Swoboda	07/27/09	Revised Draft after 3 rd Beta test	0.4 draft 4
Rob Swoboda	02/07/10	Revised Draft after 4 th for StarEast Presentation	Version 1.0

1. Abstract

This white paper defines Beta Testing and its role and importance in the Systems Development Life Cycle (SDLC). It includes an overview of the SDLC, Beta Testing and best practices for executing the Beta Testing phase of the SDLC.

In addition, this document describes how to apply the high-level SDLC concepts presented here to client /server and web based products. Associated template artifacts needed to implement the Beta Testing phase will be provided separately.

2. Systems Development Life Cycle

The Systems Development Life Cycle (SDLC) is a conceptual model used in project management to describe the phases involved in an information system development project. The SDLC extends from the initial feasibility study through maintenance of the completed application.

There are many flavors of SDLC methodology, and the number of phases can range from four to ten. The Waterfall SDLC methodology is the original SDLC methodology and the simplest to explain. The Waterfall methodology, where one phase of development feeds or falls into the next, is probably the clearest example of an SDLC so we will use it as our point of reference. Other methodologies grew out of the Waterfall model. They include Spiral, Build and Fix, Rapid Prototyping, Incremental, Rapid Application Development (RAD), Joint Application Development (JAD), Fountain Model, and the Synchronize and Stabilize Model. Sometimes, parts of several models are combined into a hybrid methodology for a specific product to address its unique needs.

Regardless of the model used, the factors generally considered most important for the success of a project are the documentation, deliverables, and artifacts for each phase of the process and how closely the plan is followed.

The Waterfall SDLC methodology can be defined as the following seven phases.

- 1) **Project Planning and Feasibility:** Establishes a high-level view of the intended project and determines its goals. A feasibility study determines whether the project should get the go-ahead. If the project is approved, the feasibility study is used to produce a project plan and budget estimates for future stages of development.
- 2) **Systems Analysis and Requirements Definition:** Gathers the requirements for the proposed system, which includes a detailed study of the business needs of the organization. Project goals are turned into specified functions, and the intended operation of the application is defined. This phase analyzes end-user information needs and creates a project timetable.
- 3) **Systems Design:** Describes necessary and nice-to-have features and operations in detail, including screen layouts, business rules, process diagrams, pseudo code, and other documentation including user interface and data design. At this phase, the overall structure is defined and the underlying logic of the product is developed.

Note: Getting the analysis, requirements and design aspect correct is critical to the success of any software product. If flaws are not caught in the Systems Design phase, they can be very expensive to resolve in the later phases of development.

- 4) **Development-Design Implementation:** In this phase, the designs are translated into code.

- 5) **Integration and Testing:** In this phase, the system is tested in a Quality Assurance test environment, which checks for errors, bugs, and interoperability. Typically, programs are written as a series of individual modules, which receive separate and detailed testing. The system is then tested as a whole, and the separate modules are brought together and tested as a complete system. Testing ensures that the interfaces between modules work well (integration testing), the system works on the intended platform with the expected volume of data (volume testing), and the system interacts correctly with other software (systems testing/ interoperability).
- 6) **Implementation:** This phase includes final verification that the system does what the user requires and has met the product requirements specification (acceptance/beta testing). All Installation and User documentation should be verified and finalized. Final verification can consist of all or a combination of the following: Internal Acceptance Testing, Installation Testing, Beta Testing, Release Acceptance Testing, and Product Deployment. In this final stage of development, the software is put into production and runs in actual business or is released to market.
- 7) **Maintenance:** This phase is continuous throughout the rest of the software's life. Changes, correction, additions, moves to a different computing platform, and ongoing product support are all part of maintenance. Requested changes and new functionality are submitted to product management, and if a new release is needed, it goes through the same SDLC process. Maintenance is the least glamorous but perhaps the most important phase, and it continues until the product reaches the end of its life. At that time, it is may be replaced with a new product or simply no longer supported.

References:

- <http://www.computerworld.com/developmenttopics/development/story/0,10801,71151,00.html>
- <http://www.startvbdotnet.com/sdlc/sdlc.aspx>

Throughout the product's life cycle, communication is essential. Each stakeholder can have a different perception of what the product should be, and lack of communication or unclear communication may result in a product that is not what the customer wanted and in the worst case, not what they needed.

For an example of break-down in process adherence and poor communication effects on the SDLC, see: <http://www.youtube.com/watch?v=OfgfnZZdMII>

3. Beta Testing

This section describes the Beta Testing phase of the SDLC. It defines Beta test terms. It also lists Best Practices and a recommended structure for a Beta test.

3.1. Beta Test Definitions

There are several definitions of Beta testing; see the sources noted in the following list.

Definition 1) expresses the key difference between Beta and all other types of testing: it is generally done by the end user or customer on their equipment with little or no direct supervision or oversight from the other product stakeholders.

- 1) **Beta Testing:** Testing of a release of a software product conducted by customers. (from <http://www.aptest.com/glossary.html>)
- 2) **Testing, beta:** (1) (Pressman) Acceptance testing performed by the customer in a live application of the software, at one or more end user sites, in an environment not controlled by the developer. (2) For medical device software such use may require an Investigational Device Exemption [IDE] or Institutional Review Board [IRB] approval. (from *Glossary of Computerized System and Software Development Terminology*)

- 3) **Beta:** Beta is the test period during which the product should be of "FCS (First Customer Ship) quality" (it is complete and usable in a production environment). The purpose of the Beta ship and test period is to test the company's ability to deliver and support the product (and not to test the product itself). Beta also serves as a chance to get a final "vote of confidence" from a few customers to help validate our own belief that the product is now ready for volume shipment to all customers.
FCS: (First Customer Ship)
FCS is the period which signifies entry into the final phase of a project. At this point, the product is considered wholly complete and ready for purchase and usage by the customers. (from <http://www.sgatester.com/softwarelife/index.htm>)
- 4) **Beta Release:** The software package is now installed in one or more beta release sites. Here the software will again be used for its intended purpose. It may now be trusted enough to be used in a production environment, but it is still not ready for widespread release to the full customer base. Shortcomings will be documented and corrected. The deliverables are a Beta Release Plan, including Beta Release Documentation, and a Beta Release Report. The Beta Release Plan, reports and analyses and accompanying information/documentation qualify as protectable trade secrets. (from <http://my.execpc.com/~mhalign/code.html>)

3.2. Beta Best Practices

The following are some suggestions to ensure a successful and productive Beta.

- 1) Control communications and perception of the Beta (single message).
- 2) Always present the Beta in the best light. If there are serious issues, don't reinforce the customer's negative perception by commiserating with them. Instead, assure them that the issue will be submitted and resolved in accordance with the processes we have in place to deal with such issues. Also, remind them that we do expect to find issues in the Beta and are prepared for them.
- 3) Set clear, achievable goals for your Beta. It is possible to start your Beta Plan as soon as the requirements for a release are finalized, but be prepared to revise this document after acceptance testing confirms exactly what the release contains.
- 4) Limit the length of time for Beta. Three months typically is an adequate amount of time to verify all items released in the Beta, collect meaningful data, and ensure that no regression issues have been introduced, using weekly reports.
- 5) It is crucial to limit the number of Beta participants to a manageable number. Consider how many Beta users and sites your Beta team can support in a worst case scenario. Too many sites may result in a greater volume of data than can be processed into meaningful data. You also will increase the odds of receiving data irrelevant to the Beta release. When you limit the number of participants, you need to be careful to select the right Beta participants.
- 6) Schedule reporting intervals to help the Beta participant focus on day-to-day occurrences of issues. It encourages them not wait until the end of Beta to document all the issues they can remember.
- 7) Choosing effective Beta sites and Beta partners is essential to a successful Beta. Key considerations in Beta participant selection include the following:
 - a) **Value:** What test items or unique configuration or conditions does the Beta site/user deliver?
 - b) **Commitment:** Ensure the Beta participant's commitment and buy-in of all involved parties at the Beta site. Beta participants should have a sincere commitment to the Beta process and the improvement of the product.
 - c) **Ability:** Assess the Beta participants' ability to provide meaningful data and feedback about the Beta release.

- d) **Attitude/Motivation:** Last but not least, consider attitude. A Beta participant may be committed to the process but still be motivated by factors that are not conducive to a fair evaluation of the product. For example, participants who are motivated by the chance to receive incentives or who pressure you to get your product to market may not give you unbiased feedback; they want it released for their own reasons. Conversely, if a Beta participant has serious issues with the current version of the product and expects the Beta release to solve all of their problems except world peace, they can have only a negative experience even if all the promised improvements are verified. Also don't run your Betas at sites where your Beta participants' job may depend on the success of the Beta release or where the participants are already unhappy or upset with your product. And, never use a site that has sued your company before for any reason!
- e) If the last two conditions are not met, very little of value will be produced from the Beta sites. One site with "an attitude issue" can destroy an otherwise perfect Beta release.
- 8) Ask qualified Beta candidates to apply for the Beta testing program. This method gauges their commitment and enhances their buy-in. This commitment must be generated and maintained through the Beta process. It can be accomplished largely through consistent, scheduled communication, including pep talks or incentives if needed. Keep up their enthusiasm!
- 9) Beta cycles of less than eight weeks generally are not very effective. Too short a Beta cycle does not provide an adequate amount of time to collect meaningful data on all items released in the Beta and ensure no regression issues have been introduced. Longer cycles are more cost effective. The initial setup and coordination can be expensive but once underway, the cost of running a four week Beta or eight one can be negligible.
- 10) Plan to release new builds to Beta participants once every two weeks to address identified issues. Try to have at least four releases per Beta. If no issues are encountered, these build cycles can be used to add low-impact (from a test and risk perspective) enhancements.
- 11) If you add a new feature or major fix to existing functionality, even a small one, during the beta process, the clock goes back to the beginning of the Beta schedule (12 weeks, for instance) and you need another three to four releases.
- 12) Incentives are not required but they are a great way of showing appreciation for your Beta partners. They can help to smooth any feathers that may have been ruffled during the Beta. They also may make the Beta partner focus more on the positive aspects when speaking about your product and the pending product release. However, any incentive must be earned through diligent participation in the Beta process. What constitutes diligent participation should be spelled out in the Beta contract or agreement. This document should also specify your company's obligation to support the Beta participant. Such an agreement must be flexible to accommodate any special considerations the Beta partner may have. Let's say that the general agreement specifies the product must be run for 40 hours a week. One site was selected because of its unique characteristics that no other candidate site possesses, but it can run the product only 25 hours a week. In this case, it would be best to lower the requirement for this site. The amendment to the agreement must be made and signed off by both parties before allowing this Beta partner access to the Beta release.
- 13) The number of required testers may vary, depending on the scope of the Beta release and the available support personnel.
- 14) Some estimates report that only 20% of Beta users send back useful beta information. However, this estimate is for end users of a general product such as MapQuest. The number can increase to 100% if the Beta partners are chosen and supported with care.
- 15) Most Beta testers will try the program when they first get it, and then lose interest. You can split your Beta population into four groups and for each new release, add another group that gets the software, so there are new Beta testers for each milestone. This approach works well, but it is not always possible to train and support four groups of Beta partners. Present each new release as a build that needs the same level of commitment as the first release they received. Also, focus their attention on the new or fixed items and any functionality relating to these items as well as general regression issues.

16) Most Information Technology personnel focus primarily on the technical aspects of the Beta release. It also is important that participants are aware of marketing concerns. To this end, any message or communications about the Beta product should go through the person directing the beta as well as the product manager. At all times, the Beta team members and stakeholders must present the Beta product in the best possible light and ensure the Beta partner that any issues encountered are being addressed according to the established process. The Beta director will distribute pertinent information to the concerned stakeholders once the product manager has approved the information for distribution.

References:

Top Twelve Tips for Running a Beta Test

Beta Test Methodology

By Joel Spolsky

Tuesday, March 02, 2004

3.3. Beta Structure

When preparing for a product's Beta phase in the SDLC, address the following items.

- 1) Define goals and responsibilities – Beta Plan document
- 2) Structure the timetable – Beta Matrix
- 3) Identify Beta sites and onsite Beta partners responsible for day-to-day Beta operations (customers)
- 4) Determine internal and external participants – Beta Matrix and reporting/meeting schedules
- 5) Create data collection and associated Beta documentation – Various documentation/communication formats.
- 6) Compile data into final version and complete Beta Report
- 7) Hold a post-release lessons learned meeting. Results should be documented and referenced when planning the next Beta release.

4. Was the Beta Successful?

How do you measure the success or failure of your Beta experience?

- If the Beta has successfully satisfied all of the items within the Beta Plan's scope of work, you can claim success.
- If the Beta has not successfully satisfied all of the items within the Beta Plan's scope of work, but you found and documented all of the issues that caused items to fail, you can claim success.

In either of these cases, the product may not go to market. If this occurs, it does not tarnish the value or effectiveness of the Beta effort. Rather, it indicates that the Beta provided valuable and substantive data to the decision makers involved in product development, which enabled them to make the right decision for this product, company, and customers. Although this outcome is not what the folks involved were hoping for, it can be the best example of a successful Beta phase.

Other factors that indicate success are how well you built the Beta team, the documentation chain, and how you handled support, reporting, and communications among all stakeholders. These factors can be assessed throughout the Beta process and must be included in the Beta Report, which is reviewed in a lessons learned meeting after product release.

5. Conclusion

Success or failure of a Beta must be determined by the accuracy of the Beta objectives as well as the quality of communications to all stakeholders.

A failed Beta can be detrimental to both your company and your company's clients. What constitutes failure of a Beta varies, and a failed Beta may look similar to a successful one. For example, you may have met all of your objectives detailed in the Beta Plan, but failed to communicate effectively with your Beta participants and the Beta team. In this case, you could verify that everything delivered in the Beta release functioned correctly but still miss feedback that the product did not meet the users' needs. Then, product management would deploy a product that the clients can't use and won't buy. In the worst case, if you failed to include measurable objectives for regression or roll-back, you may deploy a product that is extremely hard to support or that will fail and destroy the user's data.