centercode

100 Tips for Better Beta Tests

An eBook by Centercode (and friends)

v1.3

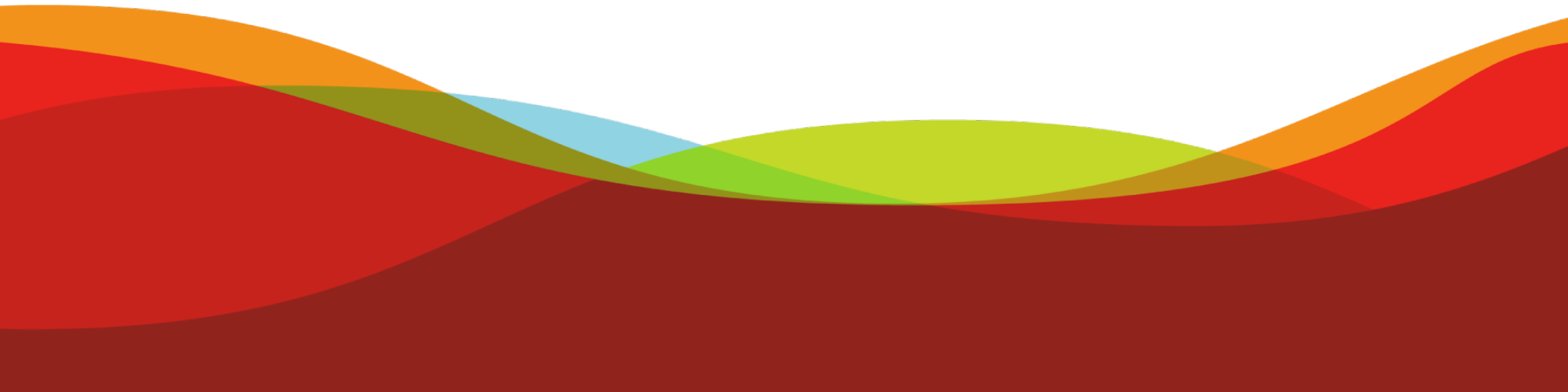
Introduction

Managing a beta test isn't easy. When you're brand new to the job, you quickly realize that there aren't a lot of standards or resources to guide you. This means that your approach will likely be forged through trial and error. And, even after running many successful tests, there will still be smart ideas you haven't adopted because the pressures of product release don't afford you much time to experiment.

We've developed this eBook to counter those challenges. Ultimately, it's part of a broader effort to distill the fundamentals of highly effective beta test management into freely available resources. However, with an effort that big, it's important to start with the basics.

We began by collecting some of our own best practices from over a decade of providing beta test management solutions to companies of all sizes; developing technology of all types. Then, we realized how helpful it would be to add some additional perspectives. Our clients typically approach beta tests, UATs, and CATs in ways that are often as diverse as the products they're creating, and they all have strengths that others may learn from. We were lucky enough to get input from many of our clients, including Adobe, Autodesk, Avid, TiVo, Symantec, and UPS. All fantastic companies with extraordinary people leading their testing efforts.

Our hope is that this compilation of tips and tricks will help you become a more knowledgeable, productive, and confident beta test manager. Whether this is your first test or fiftieth, we think that we've covered enough material that you'll surely find something of value here. Although, in case we did miss something, please feel free to let us know via comments on our blog or email tips@centercode.com.



Basic Terms

Before we get started, we'd like to clarify a few basic terms used throughout this eBook.

BETA TEST

When we say beta tests, we're using the term generally. We just mean a test where you introduce your product to a group of people who are similar to your target customers and will use your product in their real-world environments. These tests go by many names, including beta testing, field trials, pre-release, customer validation (CV), customer acceptance testing (CAT), and user acceptance testing (UAT). To many companies, these tests have subtle but important differences. However, they all share the basic idea we defined above. So, for simplicity's sake, we're using "beta" to refer to all of them.

BETA PROGRAM

When we use the term beta program, we're referring to the collection of beta tests managed at one company, whether that means beta projects for multiple products or past and future revisions of a single product. Generally, when referring to a beta program, we're discussing topics with a scope that's broader than a single test.

APPLICANTS OR CANDIDATES

We use these terms interchangeably to indicate users who have demonstrated an interest in participating in your beta test or program, but have not yet been selected as beta testers. Similarly, we refer to **alternates** as those customers who met the qualifications, but were not initially included in the primary beta test team.

BETA TESTER OR PARTICIPANT

We also use these terms interchangeably, referring to users (generally representing your actual target customers) who were selected to participate in your beta tests. We refer to the collection of all beta testers within a project as the **tester team**.

BETA SUPPORT TEAM

This term refers to your internal support team for the beta, which may be no more than a single product manager or quality engineer, or might include an entire team of stakeholders from a multitude of organizations. Generally, they're the people reviewing and responding to feedback and communicating actively with participants throughout the test.

Tip Outline

1. Planning Your Beta Test 1-12
2. Recruiting Beta Candidates 13-24
3. Selecting Beta Testers 25-32
4. Handling NDAs and Agreements 35-38
5. Kicking Off Your Beta 39-42
6. Assigning Tester Activities 43-51
7. Communicating With Testers 52-64
8. Maintaining Participation 65-75
9. Handling Feedback 76-81
10. Dealing With Scheduling 82-85
11. Incorporating Other Teams 86-90
12. Closing the Beta 91-94
13. Rewarding Your Testers 95-100



Planning Your Beta Test

1. START WITH A PROJECT PLAN

Like any well-run project, having a solid plan before you start is key. A good beta project plan includes (1) the objectives of the test; (2) target market details (i.e., beta candidate criteria) including their demographic, technical, experience, and geographic requirements; (3) beta tester participation methodology and expectations; (4) the test schedule (including planned build releases, time frames, etc.); (5) the intended size of the tester team (broken down by market segment); and (6) a list of stakeholders and their responsibilities. Other details are great, but these are all essential.

2. SET REALISTIC GOALS

There may be many goals you want to accomplish during your beta, like stressing certain features or testing different teams and resources under live customer action. However, you can only move so many mountains during a single project. If you think of each beta goal as a mini project that requires scarce resources like time and the focus of your tester team, you'll begin to understand why it's important to space things out. Generally, we recommend specifying one named goal per week, in addition to basic test functions like validating quality and collecting general product feedback. If you attempt to accomplish several major goals in tandem, you risk making little progress with any of them.

3. BALANCE YOUR CORE PARAMETERS

There are three core "moving parts" in every beta test: (1) the size of your beta tester team, (2) the duration of your beta test, and (3) the set of specific goals that you're trying to achieve. It's useful to think of these resources in equilibrium, where an adjustment to one has a countervailing effect on the others. Use this to your advantage in planning the most effective test. For example, increasing your test duration will allow you to accomplish more goals. If your schedule gets cut, you can often compensate by adding more testers and still achieve your goals. Factor these three parameters into your planning, but also keep them in mind when unexpected events require you to make adjustments throughout your beta.

4. EXPECT LAUNCH LAG

Your testers lead busy lives, and they often won't be ready to start testing immediately upon receiving the product. For software, they may wait a day or two to install. If it's a hardware product, they may not be home to receive the shipment upon delivery. There are a variety of reasons that launch lag happens, but the point is that it does happen. To compensate, it's useful to add a week or so to your plan.

5. SIZE YOUR TESTER TEAM BASED ON YOUR TARGET MARKETS

Most beta tests introduce a product to numerous target markets (or market segments), typically based on attributes such as region, gender, income, and technical knowledge or requirements. It's important to keep in mind that the number of market segments you need to reach should directly increase the size of your tester team. You don't want to work in the other direction and select a number of testers to recruit, then hope you've adequately covered your target market. If the composition of your tester team doesn't bear an accurate relationship to desired market segments, it's difficult to weigh the relevance or importance of survey results (i.e., they become anecdotal).

6. BASE PROJECT LENGTH ON GOALS

As a baseline, your beta test should be no shorter than two weeks (3-4 is generally optimal). Beyond that, the length of your beta should be tied to your project goals (and to some degree the complexity of your product). We've already discussed that under most circumstances, you should be pursuing one specific goal per week. Thus, if you have four primary goals that you want to accomplish, your beta test should be at least four weeks long. If you need to achieve more goals in a shorter period, consider increasing the size of your tester team and splitting your tester team into focused groups. That way, you're maintaining equilibrium among your core parameters, and your tester team's attention isn't being diluted by trying to address several simultaneous goals.

7. GET BUY IN

Beta tests are rarely managed by one person alone, and the data you collect almost always affects several different people and/or teams. Thus, your plan should also contain information about key stakeholders so that they're aware of their responsibilities throughout the project, key milestones, and general process descriptions. This generally includes product management, QA, and support at the very least, but may also include product marketing, sales, and members of the executive team.

8. DON'T FORGET ABOUT RAMP-UP TIME

If you're starting a beta program from scratch, recruiting a great tester team can easily take two weeks or more, depending on your target market requirements and the size of your test. If you're starting a beta project with either an existing (hopefully interested) customer list or an established beta community, ramp-up can be reduced to only a few days. Either way, it's important to include this period in your plan. The last thing you want is to sacrifice planned testing time to make your product release window because unanticipated recruitment delays consumed 25% of your beta schedule.



9. BUILD IN MULTIPLE PHASES

Splitting a beta test into phases (e.g., Beta 1, Beta 2) offers a number of advantages. One is the ability to slowly introduce a larger tester team, which allows you to reduce the impact of early bugs, ultimately burning out fewer testers. Another benefit, specific to hardware tests, is the ability to cover more of your target market (and their unique environments) with fewer expensive, pre-production units by redistributing hardware between phases. Note that if you require time between phases (a few days or more), communicate this clearly with your beta testers to ensure they remain aware and engaged. It's best to keep this downtime to an absolute minimum when possible.

10. PLAN FOR IDLE PARTICIPANTS

It's extremely uncommon for every beta tester to meet the goals you'll set for them. Sometimes participants are simply unmotivated (which other tips in this guide address directly), but many times other personal or business responsibilities take precedence. It's crucial to factor this into the recruitment section of your beta plan. If you've never managed a beta test before (therefore having minimal recruitment and participation management experience), you should plan to include at least two to three times the participants you consider necessary to meet your goals.

11. PLAN FOR CHANGE

While your plan is a great starting point, beta tests quite often change course rapidly and unexpectedly. You may run into show-stopping software issues that require additional phases or participants. Another group may run into an issue that delays a build by a week or more. Early feedback may change the primary goals. Be prepared to adjust as necessary, communicate changes clearly to all involved (especially changes regarding the beta test schedule or goals), and always update your plan accordingly.

12. PUT YOURSELF IN YOUR TESTERS' SHOES

"Plan. Think things through, pretending you are the beta tester. What do you need to help you test and report information back? Then 'beta' your test with a few people to make sure it works as you expect and that you are getting back the information you need to make your testing successful and worthwhile." **-Gayle Musker, UPS**

Recruiting Beta Candidates

13. BUILD A POOL OF TARGETED CANDIDATES

While public betas that anyone can join can be a useful marketing activity, traditional beta tests are better suited for a more limited audience. Rather than simply letting anyone in the test, it's best to build a large pool of applicants from which you'll select those who both best meet your criteria and exhibit the signs of great testers. Remember, just because a beta is public doesn't mean you'll get more feedback, but it does tend to make the feedback you get harder to analyze because you'll typically know much less about the testers submitting it.

14. RECRUIT MUCH MORE THAN YOU NEED

Not everyone who applies for your test will be a great candidate. Some won't meet the basic technical requirements; others won't match your target market; and some will demonstrate a lack of skill in clearly communicating (thus making your life much more difficult down the road). It's generally best to recruit at least three to five times more candidates than you actually plan to select, allowing you to choose only the most qualified beta testers. If you can get more, that's even better (we generally aim for ten times as many).

15. PROMOTE EXCLUSIVITY

It's a good idea to let testers know that not everyone will be selected. For one thing, it reduces the feelings of disappointment (and sometimes Internet rage) among those who were not selected. However, it also emphasizes the importance of participation right from the very first interaction with your beta project. If applicants know that they're signing up to become one of a limited number of participants, they'll understand that there's a higher degree of responsibility with that application.

16. START WITH AN APPLICATION

It's almost always beneficial to present interested candidates with an application survey containing 5-10 questions that they're required to complete. The goal of every question should be to help you identify the best testers for your specific product. Generally speaking, what you're intending to do is identify candidates who: match your target market; are responsive and effective communicators; pay attention to detail; and are genuinely excited about your company, product, or similar products.



17. RECRUIT WITH OPEN-ENDED QUESTIONS

One great way to increase the value of your application survey is to include a couple of open-ended questions. Asking something like, "Why do you want to beta test this product?" will act as a mini-interview process, allowing the tester to demonstrate the type of effort they're likely to put into feedback during the beta itself. These short answers can be invaluable in narrowing your pool to the most effective candidates.

18. KEEP YOUR RECRUITMENT VAGUE

You want people fresh and unbiased when they receive the product. Your recruitment messaging should entice them to sign up while hiding the details of the actual product. That way, they'll be excited and interested in exploring the product once it arrives. Furthermore, it provides you flexibility and eliminates issues with tester expectations.

19. STATE CLEAR REQUIREMENTS

If your product (or beta phase) has specific requirements (hardware, software, demographic, geographic, or experience/knowledge), make certain that they're clearly communicated to all test candidates. If essential, take extra steps (such as additional surveys or even personal phone calls) to verify that candidates meet the requirements. Recruiting testers who literally cannot participate (no matter how bad they want to) is a huge headache for everyone.

20. SET PARTICIPATION EXPECTATIONS

It's crucial to always let your testers know exactly what's expected of them in a beta test, from reporting bugs, to completing surveys, to participating in forum discussions with other participants. We find it best to first set expectations during recruitment, again after selection (at the test kickoff), and once every couple of weeks throughout the test.

21. IDENTIFY ANY COSTS

Beta testing is voluntary, and as a general rule, should incur no costs to participants. If there is a cost of ANY KIND associated with participating in the test (for example buying paper/ink for a printer being tested or data rates on a mobile phone), go out of your way to cover the costs yourself. If for some reason this is impossible, make it very clear to every tester before they sign up.

22. SET MANAGEABLE TIME COMMITMENTS

When a beta program has participation problems, it's easy to view testing time in extremes. You grow accustomed to seeing many participants give little to no time, while others are heroes who treat the beta like a full-time job. What you want to avoid is letting that mind-set carry over into your beta plan by building in significant time commitments but expecting only a few to keep them. Instead, establish a realistic commitment of time and effort from each beta tester but expect that they will satisfy it. It also doesn't hurt to plan special rewards for those that do go the extra 10 miles.

23. USE SOCIAL NETWORKS

Social networks are an awesome way to find willing participants. Twitter, Facebook, and LinkedIn all provide you with a free and simple mechanism to track down great candidates. If your company has active social media marketers, start a dialog and enlist their help. If you have to go this route alone, tools like TweetDeck, CoTweet, and HootSuite allow you to monitor conversations relevant to your product across social networks.

24. LOOK OUT FOR VIP TESTERS

"During the prospecting process, it will become evident that some customers are more likely to provide good feedback than others. A subset of the beta participants will warrant extra attention and support during the program as they are more likely than others to help you reach your targets. Identify these premier companies early. Make all internal parties in your company who are involved in the beta program aware that these special firms will be monitored closely, and if needed, given additional help and support to ensure they are successful in the beta. Assign named technical resources to these sites to better ensure their beta program success. The product management team should work especially close with these customers, visiting them or engaging in additional contact to help resolve issues that arise." -Jeff Crawford, Adobe

Selecting Beta Testers

25. BE SELECTIVE

Don't let just anyone into your beta. Public beta tests sound like a seductive idea because you don't have to spend time reviewing and selecting testers, and an open beta test has to result in more feedback, right? Except, it doesn't really work that way. Public betas are hard to manage, and you end up with a larger beta that gives you less feedback. With a public beta, people generally join out of curiosity, which is a much different mind-set than you get in closed beta tests. In addition, it offers a much wider group of potential customers an early look at an unfinished (and most likely quite buggy) product that can present a very negative impression, ultimately hurting sales.

26. LOOK FOR DETAIL AND CARE

Select candidates who complete a test application with great detail, limited spelling and grammatical errors, and a good argument for why they should be in the beta. These people are likely to deliver similar results during your project. On the other hand, PEOPLE WHO WRITE LIKE THSI SHULD B AVOIDED (unless, of course, that's your target market).

27. CONSIDER APPLICANT ENVIRONMENTS

Most technology relies on adjacent or parent hardware to function properly, like the PC that runs your software or the mobile phone that runs your app. These environments beyond your control can cause many issues and bugs for your product, and are one of the best reasons to run a beta. So, when it comes to selecting your participants, you want to know more about them than just personal characteristics. It's also important to know about the environments (we call them "test platforms") that they'll be using during the test. Ideally, you want to be testing your product in dirty environments that resemble what it will face when customers start taking it home (or to work). Then you can analyze the feedback you get in light of that test platform. This is one of the primary (and most well understood) benefits of beta that you don't get in the very controlled, quality lab environments.

28. SEEK OUT PASSION

If someone who meets your basic requirements is hounding you to get into a beta test, you should give that person a chance. Enthusiasm and persistence are two excellent traits in beta testers, and that person will likely be one of your more active participants.

29. AVOID FRIENDS AND FAMILY

The ideal beta participants are those who are unbiased and truly represent your target audience. This effectively eliminates friends and family, as well as most employees, from being great candidates. Sometimes their loyalty compels them to hold back, while other times they don't feel obligated to participate like other testers. Either way, your best move is to rely on testers that you don't already share a prior personal relationship with.

30. AVOID SALES LEADS AS TESTERS

It's common for sales to ask a beta manager to let a lead into a beta test. There are at least two problems with this, though. First, sales leads usually aren't interested in **testing** your product, but rather they're interested in **evaluating** your product. They're looking at it relative to their immediate needs, and you can't rely on their participation because the motivations are completely different. Second, beta testers are more willing to forgive bugs and other quality problems than a customer who's focused on evaluation. Besides, the last thing you want is an angry call from sales insisting that their lead didn't buy the product because it was too buggy during the beta.

31. MAKE SELECTIONS QUICKLY

The availability of your testers can change, so we recommend keeping the window between recruitment and beta tester selection very short (days, if possible). This will help ensure that those who volunteer are still excited for the project and still capable of committing the time necessary. Otherwise, you might find yourself reopening recruitment to find replacements for the testers you lost, further delaying your project.

32. DON'T TRUST EMAIL

If you identify a highly qualified candidate, but they don't respond to your invitation to test, it's likely they missed your email. Often, a simple phone call will reveal that your message was missed or ended up in a spam folder. People generally don't express interest in being in a beta test and then immediately turn around and change their minds.

33. ALWAYS KEEP A POOL OF ALTERNATES

There can be many reasons for losing a tester. Some realize they don't have the time to commit, while others have to be removed for disciplinary reasons. Whatever the reason, it's a good idea to have a pool of alternates available. You don't want to go through the hassle of reopening recruitment in the middle of your beta test. And having alternates also gives you the ability to add a few extra testers if it looks like you won't be hitting your participation goals.

34. NOTIFY SELECTED TESTERS ONLY

You may have the urge to let the applicants who weren't selected know that selection is complete. We've found that this is usually a bad idea. More often than not, those who weren't selected will react negatively, feeling snubbed because they weren't selected when they feel they're perfectly qualified. It's generally a better idea to set expectations ahead of the recruitment, with a message along the lines of "You will be notified within one week if you're selected to test."



Handling NDAs and Agreements

35. ALWAYS USE AN NDA

This may not need mentioning, since companies seem to request a non-disclosure agreement (NDA) before they'll even tell you where their restrooms are. However, in the context of beta testing, NDAs are critical. They offer you legal protection, but more to the point, executing an NDA (accompanied by a plain English explanation) reinforces the importance of beta secrecy to your tester team so hopefully you don't have to enforce it. We've published a free kit that covers both NDAs and participation agreements for beta testing, including templates and instructions, which is available at www.centercode.com/resources/nda.

36. USE A PARTICIPATION AGREEMENT

The role of participation agreements is a little less obvious than NDAs, but still very important. They'll cover things like participation expectations, receipt and return of the test product, limitations on what can be done with the product, etc. While they're binding legal documents, they're generally not something you try to enforce in court. Their value is in clearly communicating what's expected from your testers and reminding testers who stray what they've committed to. The kit mentioned in the previous tip also contains a lot of information about participation agreements, including the key elements they should cover.

37. SEND OUT NDA REMINDERS

Sometimes, especially in longer betas, it's a good idea to send out a mid-test email about the terms of the NDA agreement. We find that a good time is after the first few weeks or if you have a break in testing between phases. Basically, when enough time has passed for excited testers to innocently forget that they can't talk about the product, a gentle reminder of the NDA terms and consequences of breaking them is all it takes to protect confidentiality and avoid an unpleasant situation.

38. ENFORCE ZERO TOLERANCE

While it's always on a new beta manager's mind, beta NDA leaks are actually quite rare. That said, when it comes to confidentiality, no violation is too small. If someone leaks anything about your project, you shouldn't give an inch. Immediately remove the tester from the beta and notify the other testers of the action and penalty. While legal action is generally not pursued in beta NDA leak cases (outside of industrial espionage, violations are generally the result of accidents or ignorance), it is an option. Obviously this an uncomfortable series of events, but people need to recognize that you take the terms of the agreement very seriously. One crack in the dam is all it takes sometimes for the whole thing to come crumbling down.

Kicking Off Your Beta

39. START WITH BETA 101

It's a mistake to assume everyone knows how to effectively participate in a beta test, so start things off by sharing a set of simple guidelines that explain what makes a great tester. You can also use this opportunity to introduce testers to the systems or tools you will use to manage their feedback (providing all relevant URLs), as well as provide tips for how to write great bug reports, contribute positively in forums, etc.

40. REAFFIRM EXPECTATIONS

We mention expectations again because clearly communicated expectations are a big part of a successful beta test. Once the test is ready to start, you should send out another message letting testers know what you want, when you want it, and how you expect them to accomplish it. This allows you to establish a clear path to the incentives (i.e., if users are doing what is asked, they get the reward). It also gives you a concrete reference point to leverage in future discussions with any testers who are not participating.

41. CONFIRM CONTACT INFORMATION

Before any test, and especially before shipping any physical product, remind all participants to update their contact information in whatever system you're using. People move, change cell phone numbers, and have email addresses they only use for signing up for things. You want to make sure you have the most complete and current information available. Plus, shipping an unreleased product to the wrong house can be a source of frustration and embarrassment (not to mention a waste of time).

42. INCLUDE RETURN INSTRUCTIONS

If you're shipping physical products that you expect testers to return, be certain to include clear instructions on how to return the product and what you expect returned. Be sure to include everything necessary to ship the item (**prepaid**, of course). Skipping this last step can greatly reduce the amount of product you eventually get back. It's also helpful if testers can ship back product in the same box it arrived in. It means fewer shipping hassles for them, plus it gives them a distinct place to store any instructions and return labels you provide at the onset of testing.

Assigning Tester Activities

43. USE THE RIGHT TOOL FOR THE JOB

There are many types of feedback mechanisms. Beyond bug reports and feature suggestions, other common activities in betas include the use of surveys, tasks, daily journals, discussion forums, wikis, etc. The key is to think about your goals and select your tools based on what you're trying to accomplish. For example, if you know that documentation feedback is critical, that's a great opportunity to enter your docs into a wiki and crowdsource improvements.

44. USE GENERAL BROAD TASKS TO ENCOURAGE PARTICIPATION

Some testers lack the initial drive to independently explore your product and report back their findings. We've found that giving people a set of very basic, general tasks will help kick-start their use of the product, after which they're more likely to do their own exploration. Note that these should not include tasks that will focus the tester on very specific features or activities, but rather the product as a whole (i.e., download the software; load the software; review the online help). In most cases, while you may have to nurture participation in the beginning, testers will be much more independent once they build some momentum.

45. USE SPECIFIC TASKS TO REGRESS FIXES

One area where a diverse and reliable tester team really shines is in regression testing. If you've fixed some known bugs, verify that you've solved the problem with a group (or, in some cases, all) of your testers. You can segment your team by test platforms that were known to exhibit the bug and assign tasks that follow the specific steps required to recreate the issue. Or, you can set your entire team after the problem just to make sure it's really gone. The added benefit is that testers will experience the results of their efforts firsthand, leading to increased participation.

46. ASSIGN OBJECTIVES, NOT STEPS

"Goal-based test scripts are a much better way to get a thorough assessment of your software's usability. If you give them a task like 'Turn on the Scheduler,' you not only assess how the Scheduler works, but how easy it is to find and use it." -Tony Weiss, Symantec

47. GIVE REASONABLE DEADLINES

Remember, the people volunteering to test your product are (usually) doing this in their free time. It's important to respect that and give them a real opportunity to finish the activities (surveys, tasks, etc.) assigned to them. We find that 2-3 days is reasonable for relatively simple tasks, while a week is appropriate for more complex assignments. You can opt for shorter deadlines when necessary (and only sparingly), but understand that completion rates will probably suffer for it.

48. ISSUE SURVEYS OCCASIONALLY

Try to limit surveys to about one or two per week. They're an incredibly useful tool in beta management, but there are consequences to assigning too many. Frequent assignments discourage your participants from exploring the product on their own, and if taken to the extreme, they will quickly frustrate and burn out your testers.

49. AVOID TESTING FAILURES & LOST CAUSES

"Don't ask testers to perform a test you know will fail. This just causes unneeded frustration. Similarly, don't ask your testers a question unless you can act on the feedback. Testers want to make a difference (which is why they volunteered to test), so if their feedback goes unheard, they will be upset and will be less likely to volunteer for you in the future."
-Amanda Dawson, TiVo

50. LIMIT YOUR SURVEY QUESTIONS

When testers are presented with a long survey, they may decide to skip it based on looks alone. It's not necessarily important whether the questions are simple or complex. Thus, we recommend limiting surveys to between 5 and 15 questions. An even better approach is to only display the most pertinent questions by making them conditional based on their answers to previous questions (a feature found in our software). Not only does this give you the opportunity to craft a more detailed survey, but it also makes the survey initially appear very short.

51. TIME ASSIGNMENTS STRATEGICALLY

If you're planning a long beta, it can be helpful to time your tasks and surveys so they occur in the later stages of testing. Early on, many testers are likely to be excited and focused on exploring, not needing the additional push. However, as the test goes on, the extra direction that assignments provide can often breathe new life into a waning beta. Tasks make it very clear what you are looking for from the tester, while surveys tend to be an easy method of giving feedback (particularly when they consist mostly of drop-downs, check boxes, and radio buttons).

Communicating With Testers

52. USE EMAIL SPARINGLY

You'll want to ensure email communication is infrequent, important, and to the point. Consider limiting its use to messages that are either time -sensitive or very important. For other matters, stick with your beta test tools (bug report comments, discussion forums, etc.) as your primary means of communication. Why? High email volume risks frustrating your testers and diluting the perceived importance of your messages. Sending frequent emails also blurs the lines of your communication protocol, encouraging testers to direct communication to you via email instead of your beta test tools.

53. OFFER POSITIVE REINFORCEMENT

Don't forget to praise users for particularly good feedback or participation in the test. Comments like "excellent question" or "great thought" not only make your users feel appreciated, they also reinforce that you'd like to see more of that behavior. After all, it's sometimes easy to overestimate how much guidance you've actually given your testers.

54. GIVE PREFERENTIAL TREATMENT

"Pay extra attention to your super users. Some testers are like part-time employees and will spend hours each day testing, helping others in the forums, and evangelizing your product. Reach out to these testers with a personal email, phone call, or extra gifts to make them feel special. It's worth it in the end!" -Geoff Griffin, TiVo

55. ENFORCE COMMUNICATION PROTOCOL

If you want to keep your beta test data organized, enforcing communication protocol is critical. When participants have general test concerns, like a change in their availability or questions about how to be a better tester, it's okay for that communication to happen by any means necessary. Where you want to be strict is in how participants give you beta test feedback. If some people use your beta management tools, others send in random emails, and one person calls you to report bugs, things can get out of control quickly.



56. WRITE FOR THE APPROPRIATE AUDIENCE

It's important to remember your audience demographics when communicating with participants. Take young adult gamers as an example. They'll tend to speak, engage, and operate a lot differently than older business software testers. That doesn't mean your tasks have to revolve around frags and pwning, but don't expect them to provide detailed cost-benefit analyses for your product.

57. TIME YOUR COMMUNICATIONS

"If you are running a global program, or even testing across country, be mindful of when you send the email based on time zones. Depending on the target audience, you may find they have different habits as to when they check email and when they arrive at/depart from work. Don't be afraid to ask these kinds of questions of your beta participants in exit surveys, and remember that you are asking them to help you free of charge in most cases, so you need to flex to their schedule, not the other way around. Scheduling outbound emails via macros based on time zones can yield a high return rate if done right." -James McKey, Symantec

58. CLICK SEND

"Don't be paralyzed by perfection with those emails you have to send to lots of beta participants. You'll have to send several of them to establish good communication and follow up with your external beta testers. If you brood over the perfect language rather than just using your own casual, engaging tone (assuming you have one), then you'll likely communicate too little, too late, or in a manner that puts off your audience." -James McKey, Symantec

59. STICK TO YOUR WORD

Keep your promises and choose your words carefully. If you say something during a beta test, your testers will hold you to it. This applies to all sorts of issues, including test schedules, product features, and (obviously) incentives. You might think testers won't care if you have to go back on something you've said, but you'd be underestimating the personal stake they feel when it comes to testing your product.

60. KEEP THINGS UNDER WRAPS

Now that you know how important it is to watch what you say, what do you do when testers ask hard questions? You still have the option to not say anything. The point of a beta project is to gather tester feedback, so as long as you aren't ignoring the question or otherwise being rude, you aren't obligated to give full answers or any answer at all. The key is to handle it in such a way that you don't upset the tester and derail future participation.

61. DON'T ANNOUNCE UPCOMING BUILDS

This one is a cardinal rule of beta testing. Never, ever tell participants that you have a new build coming. They will stop testing and wait to see what the new build brings if they know a release is on its way. And if you think about it from their perspective, that makes sense. Why would they keep testing when anything they uncover could already be fixed unbeknownst to them? You see it differently because you probably have the benefit of knowing what is being addressed in the new build.

62. BE PROFESSIONAL, BUT BE HUMAN TOO

Balance professionalism with personality when managing your tests. Not only will this help your participants feel like they are part of a team, but it can also pay dividends when it comes to participation levels. It's hard for testers to relate to a stodgy and stilted beta manager, and if they can't relate to you they'll feel less compelled to help you.

63. DON'T BURN OUT GOOD TESTERS

It's natural to want to identify good testers and keep inviting them to more tests. However, even active and engaged beta testers can suffer burn out. Keep a careful watch on their productivity if you use them often, so you know when this might be happening. And if you want to keep your relationship going with the tester while still imposing a break, consider using them as an alternate.

64. CONSIDER HOME OR OFFICE VISITS

"Try arranging visits with beta customers in their test environment (often, if you can). Invite product designers along for the trip, too. One of the challenges of product development is verifying assumptions about how your users use your product. What better way is there to do that than seeing for yourself?" -William Marshall, Avid

Maintaining Participation

65. USE GOOD TOOLS

At the risk of self-promoting, beta tools matter. Giving beta testers access to good tools for sharing results helps ensure that you actually get feedback. Tester-friendly and well-organized beta tools build momentum early in the test. Whereas, if your testers struggle to share data with you or have to juggle a bunch of different tools (and separate user logins), they're less likely to participate. There's a reason why we see an average of 90% participation in our beta tests, and the Centercode platform is a big part of that. Beyond reducing tester participation, a collection of disconnected tools also makes it much more difficult to monitor participation as a whole, as well as work with the incoming feedback.

66. THROW OUT THE BAD APPLE

One bad tester can spoil the whole bunch. If you have a person who is excessively abusive, negative, or offensive, it's important that you pull them from the test. Otherwise, you risk letting that person ruin the experience for your other testers. The more people are negatively affected by a bad tester, the less likely they are to participate. So, effectively, you're not only hurting your other testers but your product as well.

67. RELY ON OPEN-ENDED QUESTIONS

You can really encourage participation by relying on open-ended questions. When you ask simple "yes" or "no" questions, you're closing off opportunities to learn more about what participants think of the product. They might have had great feedback if the question asked "why" or "how" instead of "yes" or "no". There are times when you might have to use simple, closed-ended questions — particularly toward the end of long beta tests, when participation is dragging — but these types of questions should be a fallback, not a staple.

68. USE EXPEDITED SHIPPING

If you're testing hardware, use overnight shipping if possible. The added expense comes with benefits. First, it'll help you ramp up the test faster. Everyone experiences launch lag, but when your product spends less time in transit, you get shorter test phases. Second, it sends a very positive message to your testers. By spending more on overnight shipping, your testers will see that you're just as eager as they are to start testing the product.

69. PRACTICE CREATIVE THANKFULNESS

We always encourage you to thank your testers early and often. They really respond when you show them you appreciate their feedback. By offering positive reinforcement throughout and incentives at the end of your beta, you'll be well on your way to making your testers happy. If you want to build an even more positive relationship, though, consider adding something creative, unique, and commemorative into the mix. Tester Team T-shirts, for example, show your appreciation and add an additional sense of exclusivity to projects.

70. BE A GOOD MODERATOR

This means being careful to watch for tangents and diversions. They're a natural part of beta testing, especially when you run discussion forums for your testers. But if testers start to focus on one thing too long or become mired in unproductive discussions, don't be afraid to gently steer them back to the charted course.

71. EMPHASIZE EXCLUSIVITY

One of the best tools at your disposal for building a sense of community is the exclusivity of beta testing. Testers thrive on that exclusivity. It makes them feel like they are themselves special as well as part of something special. The result is a more dynamic community of beta testers that are each more motivated to help shape your product.

72. AVOID OVER-RELEASING

It can be a fine line, but try to keep product updates to a minimum during your beta. On one hand, updating your product during beta does show responsiveness and gives you a chance to perform regression testing. On the other hand, too many updates can frustrate your testers and discourage participation. Even simple software installations can get tedious if you subject your testers to them regularly (hardware can be much worse). And if you release updates frequently enough to become predictable, people will stop testing in much the same way as if you were announcing upcoming builds.

73. DON'T DEMAND CONFORMITY

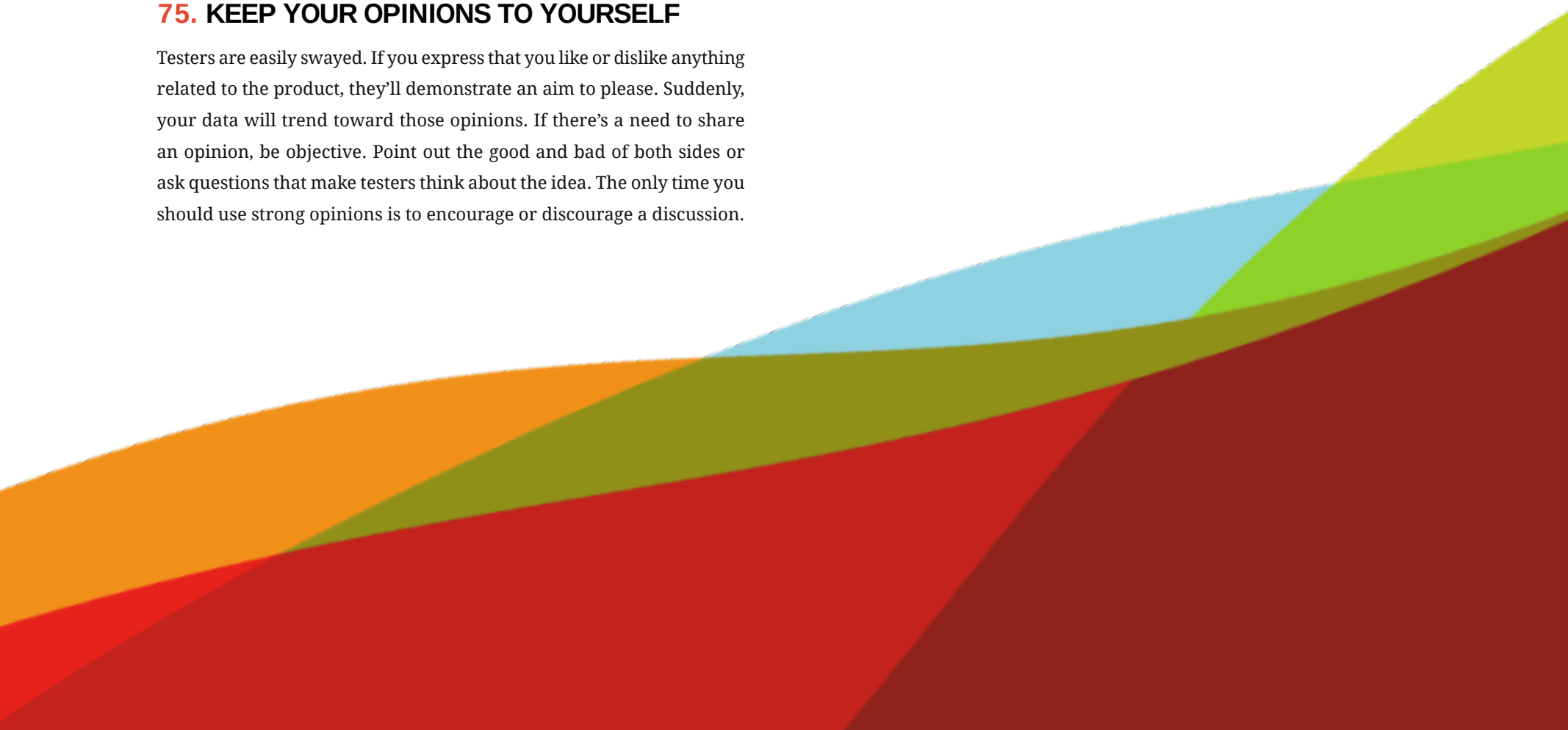
Beta testers are all unique. Sometimes it's better to adjust the test to accommodate them than to force them to comply with specific demands. For example, while user forums are a great way to keep users invested daily, some people just aren't social enough to effectively utilize that channel. In that case, asking for daily update journals are a better option that produces the same results.

74. ACT QUICKLY WITH IDLE TESTERS

The longer you let idle testers slide, the more it seems that you're not serious about participation requirements. Give your testers a gentle nudge if you notice they haven't been providing feedback regularly. A simple email showing them what they've done and what you expect will get most people engaged. If you let it linger, they will think you don't care and then it will be too late.

75. KEEP YOUR OPINIONS TO YOURSELF

Testers are easily swayed. If you express that you like or dislike anything related to the product, they'll demonstrate an aim to please. Suddenly, your data will trend toward those opinions. If there's a need to share an opinion, be objective. Point out the good and bad of both sides or ask questions that make testers think about the idea. The only time you should use strong opinions is to encourage or discourage a discussion.



Handling Feedback

76. BE RESPONSIVE

One of the easiest ways to improve participation in your beta is to be responsive. Participants understandably get discouraged when they offer feedback but receive no response or even an indication it has been read. By being responsive, you indicate that their feedback is valued. It's also an opportunity for you to get more information on reported issues.

77. BE GENUINE AND SPECIFIC

There's a caveat to being more responsive: You should never fake it with boilerplate responses to testers' feedback. Giving the same canned response to all feedback is hardly better than no response at all. Users see it is impersonal and it makes them feel like you don't really care about their input. Instead, make your response specific to the feedback at hand or the user's participation in general (e.g. "Thanks for your hard work recently. You've given us some very helpful bug reports and feature ideas this week.")

78. USE BUG FREQUENCY FOR SEVERITY

Redundant beta feedback may sound like a nuisance, but it's actually an excellent way to measure the severity of bugs. If a significant number of testers report the same problem, this immediately escalates its severity. Thus, it's important to encourage your participants to report all problems they encounter. Doing triage on bug reports is a much better problem to have than releasing a product with small, widespread bugs that slipped through the cracks during beta.

79. USE A VARIETY OF FEEDBACK MECHANISMS

It's good practice to offer your testers several ways to share feedback. Some people like discussing the product on forums where they can chat with others. Other people aren't comfortable with the idea of bug reports, but will provide amazingly detailed feedback in a daily testing journal. Some thrive when given specific directions via tasks and surveys. By providing different opportunities to participate, your testers are more likely to find feedback mechanisms that resonate with their individual preferences and ongoing needs.



80. STAY OFF THE LOW ROAD

At some point, you'll run across a participant who doesn't like your product, gets frustrated with a bug, or just has a negative attitude. This can be a tough situation, but try not to suppress criticism or respond negatively to it in any way. If you publicly censor testers, you risk alienating them and limiting future feedback. And if you attempt to rebut what was said, it looks like you're not interested in honest and candid feedback. On the other hand, if you feel like you could learn more about the problem by asking questions, by all means do so. Just be careful of your mindset.

81. ENABLE TESTER CREATIVITY

You might think it's best to discourage testers from using your product in unintended ways, but there are benefits as well. When you allow testers to use the product how they want, you're tapping into a great resource for future product and feature ideas. And since you can't always control how paying customers will use the product, it's also a chance to get insight into support issues that may arise from unforeseen use cases.



Dealing With Scheduling

82. BE REGIMENTED, BUT NOT SPECIFIC

It's best to be vague about dates when talking to your testers. The nature of beta testing often requires you to shift your schedule to accommodate issues and changes. If you lay out everything in numbers of weeks, most of what you've told testers will still be true if you have to adjust your schedule. If you give testers specific dates, one small hiccup cascades into a dramatic change that affects every subsequent date.

83. PRACTICE CONSISTENCY

"Keep to a regular test schedule. Overall participation improves when keeping to a consistent test schedule. I send out new testing instructions on the same day each week and ask testers to complete surveys by the same day each week. I'll often see my numbers drop off when forced to diverge from our regular schedule." -Geoff Griffin, TiVo

84. INCENTIVIZE EXTENSIONS

If you need to extend your test beyond the specified period, it's important to budget for additional incentives. Testers committed to specific terms, and your good will with them can take a hit if you extend those terms without sweetening the bargain. Just as important, the extended commitment needs to be optional. It's likely not their fault the test ran long, and they shouldn't be penalized if they can't continue.

85. MAKE THE MOST OF SLIPS

If you have already shipped a product or people are prepared to test but some last minute issue has caused the project to slip, you need to change your focus fast. Examine a part of the product that is unrelated to the slip and ask testers to focus their energy on it. If the slip is severe, it may make sense to put the test on hold. However, only do that if you believe that you can't solve the issue within a day or two of test start. It's very difficult to regain test enthusiasm and momentum once a project has been placed on hold.

Incorporating Other Teams

86. PROACTIVELY SHARE WITH OTHERS

We recommend finding time to share beta feedback with teams that might not be involved in the day-to-day efforts. For example, you can pass along feedback about documentation to your technical writers. You might also share product feedback and testimonials with your marketing department. Reaching out like this helps bolster your organization's opinion of beta testing, gives other departments an opportunity to elicit valuable feedback, and may even help your career trajectory.

87. ENGAGE WITH YOUR QUALITY TEAMS

"Make friends with your internal quality department. They will know the product very well and will be able to tell you issues that they frequently encounter, which you can in turn tell the testers. They can also tell you what's been fixed in a release." -Amanda Dawson, TiVo

88. BECOME A JACK OF ALL TRADES

"Being a beta program manager requires some well-rounded skills. You have to be good with customer facing engagement, proficient at marketing techniques, competent at managing a project, and able to handle the details of following up with all customer issues/concerns while compiling a detailed final report that can be ready for delivery very shortly after the program ends. If you feel you are weak in any of these areas, compensate in your beta plan by requesting assistance from the necessary group (marketing, sales, etc.) to fill in the gaps." -James McKey, Symantec

89. DON'T BE A FIREHOSE

Beta tests generate a wide variety of feedback, covering every aspect of your product. While product managers are often interested in the big picture, most of your colleagues only have the bandwidth for what's relevant to them. Providing too much information will limit their engagement and perception of the value of your beta. Do your best to ensure that the feedback you're distributing is valuable to those receiving it. This often means bugs for QA, survey results for marketing, feature requests for product planning, etc.

90. PROMOTE THYSELF

"Market yourself internally. The job of a beta program manager is often misunderstood and underestimated. Document what you had to do along the way and consider presenting the entire process for recording as a video presentation. At the very least this will give confidence to your team that someone else could jump in and can carry on with a rough guideline, aware of the potential pitfalls, if you were to suddenly be abducted by aliens and not returned for at least six months."
-James McKey, Symantec



Closing the Beta

91. GIVE NOTICE THAT IT'S OVER

Send a clear indication to testers that the project is complete. An email, official letter, or some other form of announcement helps bring closure to the test and lets people understand their role is complete. It's also helpful to use this time to remind testers of the terms of the NDA should they extend past the end of the beta.

92. WAIT UNTIL YOU RECEIVE THE PRODUCT

On the other hand, if participants are supposed to ship product back to you after testing, do not announce the project has closed until everything is back. If they think you have officially ended the test, they are far less likely to return the test items. They'll just think you've moved on to other, more important matters.

93. KNOW WHO'S MAKING THE DIFFERENCE

"Cultivate good testers. Good testers are hard to find and are worth numerous average testers. So be sure to keep track of the good testers and make them feel appreciated." -Amanda Dawson, TiVo

94. CREATE A SHUTDOWN PROCEDURE

"Establish a standard process for shutting down betas — and stick to it. The details are not too important, but having a process improves communication. We like to post a heads up that the beta will go read-only in 7 days and then be shut down completely after another 7 days. This gives customers time to file any unfinished feedback etc."
-Simon Bosley, Autodesk

Rewarding Your Testers

95. MAKE IT WORTH THEIR WHILE

"Provide an incentive. A gift at the end of your trial, creating a better product that testers care about, or maybe inclusion in future beta programs. Testers need a good reason for spending their time testing." -Geoff Griffin, TiVo

96. WAIT UNTIL IT'S REALLY OVER

When testers receive their incentive for participating, that signals to them that the test is over. So, naturally, if you still need data or feedback from testers, you shouldn't distribute incentives. Otherwise you might find the participation levels on those last remaining tasks a little lacking.

97. OFFER WHAT THEY WANTED ALL ALONG

The best incentive for a beta test is usually the product being tested. Users signed up to test it for a reason. That's not to say other incentives aren't appreciated — gift cards from retailers like Amazon are usually popular as well — but giving participants a non-product incentive is a little like depriving them of the fruits of their labor.

98. KEEP THE INCENTIVE SECRET

Giving out information about incentives before or during the test is risky. If your testers like the incentive too much, they might make up data or participate in other unhelpful ways to make sure they look deserving. On the other hand, if testers don't like the incentive, they may stop participating entirely.

99. REMEMBER INCENTIVES SET PRECEDENT

What you give as an incentive for one test affects how people think about your future tests. This means you have to be careful about making incentives too cheap or too expensive. If they are too cheap, testers will feel like you do not value their input. And if you make them too expensive, testers will expect that from every project they test.

100. TRY SOMETHING INTERESTING

"Don't be afraid to try enticing participants through techniques that others have failed with before you or discouraged as too risky. Most great ideas that have received positive buzz (unique contests or methods of feedback) were always invariably laughed at and seen as likely to fail before they succeeded stupendously. Such ideas are rarely easy and/or safe in planning or implementation." -James McKey, Symantec

centercode

Delta Testing Solutions

WHY CENTERCODE?

Increase Revenue
½ ☆ INCREASE
in product ratings

Release Sooner
50% FASTER
test cycles

Gain Participation
75% INCREASE
in tester engagement

Save Time
40%+ DECREASE
in time managing tests



The Centercode Platform equips you with the automation and best practices you need to get actionable user insights for any use case.

PRODUCT LAUNCH

Engage your **Target Market** to successfully bring new products and services to market

USE CASES:

- Beta Tests
- User Acceptance Tests
- Field Tests

PRODUCT GROWTH

Recruit **Existing Customers** to continuously mature live products and services

USE CASES:

- Progressive Delivery
- Early Access
- Customer Advisory Boards

CUSTOMER ZERO

Gather **Employee** feedback to enhance your products while fostering a product culture

USE CASES:

- Alpha Tests
- Dogfooding
- Employee Tests

COMPETITIVE EDGE

Enlist **Competitors' Customers** to rapidly gain actionable intel about their products

USE CASES:

- Competitive Intelligence
- Market Testing
- Comparative Research