

Mo' Beta

At some time in every product cycle, the executive team wants to help product management "improve" its customer beta process¹. This is generally because the last beta took too long, didn't get enough useful customer feedback, or failed to prime the revenue pump for a post-GA sales blitz. Notice that these three goals are mutually exclusive...



One way out of the beta dilemma is to recognize the different audiences and objectives for a beta cycle, then structure different programs for each. Here, I've sorted beta prospects into three camps: the Loyal Opposition, the Overcommitted, and the Reluctant Volunteers.

The Loyal Opposition

These are the most fanatical technologists among your current customer base. If you're big enough to have a user group, the Loyal Opposition crowds into late afternoon roadmap sessions to trade configuration tips and lobby for missing features. They love your product almost as much as they love pointing out minor errors.

If you don't already, you should pamper the Loyal Opposition. During beta testing, they will shred your pre-release software, grammar-check your documentation, and invent bizarre uses for your new features. (These are all good things!) They **don't generate any short-term incremental revenue**, though, because they are already paying customers.



Plan as many early code drops as you can to the Loyal Opposition: this supplements your internal QA effort with unconventional thinking. Consider a limited edition t-shirt for those who find bugs, and keep your sales team away from these eccentric gems. Drop compliments about them in obscure USENET locations.

The Overcommitted

These customers and prospects are easy to recognize: they call your sales team frequently for assurance that your release is still on schedule. In the extreme, they have built their own product or service delivery schedule around your GA release dates – and have no slack. When your GA delivery date inevitably slips, they are in deep water.

Including the Overcommitted in your beta program is required, not optional. It gives your sales team a face-saving way to gloss over promises, and gives customers a hope of hiding their optimism. You may grumble and refuse initially, but expect to be overruled by the VP Sales.



Overcommitted beta customers are high risk, though. They will expect production-ready products that are fully tested, since your sales team has given them this impression. Some will put your “early GA release” directly into production. If your testing cycle has been shortened to recover lost engineering time, you may have a disaster on your hands. Try to assign a support engineer or smart field SE to each account, and be prepared to demand some very quick fixes from Engineering.

Reluctant Volunteers

Let’s be realistic: most beta customers never install your product. Your handcrafted CD and installation manual will probably be shelved next to 80 other untouched beta products.



Reluctant Volunteers come from a pair of mistaken impressions: sales teams think that beta installations will help close deals faster, and CIOs think their network managers (or sys admins, or database experts, or help desk teams, or software architects) have idle time to play with new stuff. Your sales rep “calls high” and convinces the customer’s CIO to be a beta site. When this trickles down the org chart, no one offers to give up another weekend to debug your code. (If you look up “passive aggressive” in the dictionary, you’ll see a picture of a Reluctant Volunteer.)

You’ll spend most of your beta program chasing Reluctant Volunteers. Consider asking each for a test plan as a rapid way to sort the interested few from the uninvolved many.

Recapping:

Group	Result	Revenue	Risk
Loyal Opposition	Real feedback if given a long beta cycle	None	None
Overcommitted	May save face for sales rep and customer	Some	High
Reluctant Volunteers	None	None	Wasted effort

Sound Bytes

In theory, we all love beta testing. In practice, loyal customers are joined by a few panicked prospects in a rush to general release. This generates scant feedback and minimal revenue. If you want useful results, plan a long beta phase for friendly customers followed by a short, post-QA cycle for urgent situations.

"Product Bytes" is a monthly newsletter about product strategies for technology executives. To subscribe, send an email to subscribe@mironov.com. Back issues are archived at www.mironov.com.

Mironov Consulting specializes in early product strategy, technology planning, and customer requirements. Call if your development schedule is science fiction or your requirements came from a séance. All contents © 2002 by Rich Mironov. Product Bytes (TM) 2002. Contact rich@mironov.com or 650.315.7394.

¹ For those unfamiliar with beta tests, this is the phase after a (software) product is fully coded and initially tested, but not yet ready for revenue shipments. The classic development phases are called alpha (internal coding and rough testing), beta or limited release, and GA ("general availability") or FCS ("first customer shipment"). Every start-up cheats on testing in order to ship GA revenue units sooner.