83: One Month Tuneup

Issue a major update 30 days after launch

A quick update shows momentum. It shows you're listening. It shows you've got more tricks up your sleeve. It gives you a second wave of buzz. It reaffirms initial good feelings. It gives you something to talk about and others to blog about.

Knowing a quick upgrade is coming also lets you put the focus on the most crucial components before launch. Instead of trying to squeeze in a few more things, you can start by perfecting just the core feature set. Then you can "air out" the product in the real world. Once it's out there you can start getting customer feedback and you'll know which areas require attention next.

This baby-step approach worked well for Backpack. We launched the base product first and then, a few weeks later, added features like Backpack Mobile for handhelds and tagging since those things are what our customers told us they wanted most.

84: Keep the Posts Coming

Show your product is alive by keeping an ongoing product development blog post-launch

Don't stop blogging once you launch. Show your product is a living creature by keeping a dedicated blog that you update frequently (at least once a week, more often if you can).

Things to include:

- Faq
- How-tos
- Tips & tricks
- New features, updates, & fixes
- Buzz/press

A blog not only shows your app is alive, it makes your company seem more human. Again, don't be afraid to keep the tone friendly and personal. Small teams sometimes feel like they need to sound big and ultra-professional all the time. It's almost like a business version of the Napoleon Complex. Don't sweat sounding small. Revel in the fact that you can talk to customers like a friend.

It's Alive

> *A frequently-updated product blog is the best indicator that a webapp is in active development, that it's loved and that there's a light on at home. An abandoned product blog is a sign of an abandoned product, and says the people in charge are asleep at the wheel.*
>
> *Keep the conversation going with your users on your product blog, and be transparent and generous with the information you share. Let your company's philosophies shine through. Openly link and discuss competitors. Hint at upcoming features and keep comments open for feedback.*
>
> *A living product is one that's talking and listening to its users. A frequently-updated product blog promotes transparency, a sense of community and loyalty to your brand. Extra, free publicity is a bonus.*
>
> *As editor at Lifehacker, I scan the product blogs of webapps I love continuously — like Google, Flickr, Yahoo, del.icio.us, and Basecamp product blogs. I'm much more likely to mention them*

> *than webapps that send out one-sided press releases out of the blue and don't maintain an open conversation with their users and fans.*

—Gina Trapani, web developer and editor of Lifehacker, the productivity and software guide

85: Better, Not Beta

Don’t use “beta” as a scapegoat

These days it feels like everything is in beta stage forever. That’s a cop out. An interminable beta stage tells customers you’re not really committed to rolling out a finished product. It says, “Use this, but if it’s not perfect, it’s not our fault.”

Beta passes the buck to your customers. If you’re not confident enough about your release then how can you expect the public to be? Private betas are fine, public betas are bullshit. If it’s not good enough for public consumption don’t give it to the public to consume.

Don’t wait for your product to reach perfection. It’s not gonna happen. Take responsibility for what you’re releasing. Put it out and call it a release. Otherwise, you’re just making excuses.

Beta is Meaningless

> *Blame Google, et al, for causing problems like this. For now, users have been trained by the aggregate of developers to think “beta” doesn’t really mean anything.*

—Mary Hodder, information architect and interaction designer (from The Definition of Beta)

All the Time

> *Is it just me, or are we all in beta, all the time?*

—Jim Coudal, founder, Coudal Partners

86: All Bugs Are Not Created Equal

Prioritize your bugs (and even ignore some of them)

Just because you discover a bug in your product, doesn't mean it's time to panic. All software has bugs — it's just a fact of life.

You don't have to fix each bug instantly. Most bugs are annoying, not destroying. Annoyances can be tabled for a bit. Bugs that result in "it doesn't look right" errors or other misdemeanor miscues can safely be set aside for a while. If a bug destroys your database, though, you obviously need to fix it immediately.

Prioritize your bugs. How many people are affected? How bad is the problem? Does this bug deserve immediate attention or can it wait? What can you do right now that will have the greatest impact for the greatest number of people? Often times adding a new feature may even be more important to your app than fixing an existing bug.

Also, don't create a culture of fear surrounding bugs. Bugs happen. Don't constantly seek someone to blame. The last thing you want is an environment where bugs are shoved under the rug instead of openly discussed.

And remember what we said earlier about the importance of honesty. If customers complain about a bug, be straight up with them. Tell them you've noted the issue and are dealing with it. If it won't be addressed right away, explain why and say you're focusing on areas of the product that affect a greater number of people. Honesty is the best policy.

87: Ride Out the Storm

Wait until knee-jerk reactions to changes die down before taking action

When you rock the boat, there will be waves. After you introduce a new feature, change a policy, or remove something, knee-jerk reactions, often negative, will pour in.

Resist the urge to panic or rapidly change things in response. Passions flare in the beginning. But if you ride out this initial 24-48 hour period, things will usually settle down. Most people respond before they've really dug in and used whatever you've added (or gotten along with what you've removed). So sit back, take it all in, and don't make a move until some time has passed. Then you'll be able to offer a more reasoned response.

Also, remember that negative reactions are almost always louder and more passionate than positive ones. In fact, you may only hear negative voices even when the majority of your base is happy about a change. Make sure you don't foolishly backpedal on a necessary, but controversial, decision.

88: Keep Up With the Joneses

Subscribe to news feeds about your competitors

Subscribe to news feeds about both your product and your competitors (it’s always wise to know the ways of one’s enemy). Use services like PubSub, Technorati, Feedster, and others to stay up to date (for keywords, use company names and product names). With RSS, this constantly changing info will be delivered right to you so you’re always up to speed.

89: Beware the Bloat Monster

More mature doesn’t have to mean more complicated

As things progress, don’t be afraid to resist bloat. The temptation will be to scale up. But it doesn’t have to be that way. Just because something gets older and more mature, doesn’t mean it needs to get more complicated.

You don’t have to become an outer space pen that writes upside down. Sometimes it’s ok to just be a pencil. You don’t need to be a swiss-army knife. You can just be a screwdriver. You don’t need to build a diving watch that’s safe at 5,000 meters if your customers are land-lovers who just want to know what the time is.

Don’t inflate just for the sake of inflating. That’s how apps get bloated.

New doesn’t always mean improved. Sometimes there’s a point where you should just let a product be.

This is one of the key benefits to building web-based software instead of traditional desktop based software. Desktop software makers such as Adobe, Intuit, and Microsoft need to sell you new versions every year. And since they can’t just sell you the same version, they have to justify the expense by adding new features. That’s where the bloat begins.

With web-based software based on the subscription model, people pay a monthly fee to use the service. You don’t need to keep upselling them by adding more and more and more, you just need to provide an ongoing valuable service.

90: Go With the Flow

Be open to new paths and changes in direction

Part of the beauty of a web app is its fluidity. You don't wrap it up in a box, ship it, and then wait years for the next release. You can tweak and change as you go along. Be open to the fact that your original idea may not be your best one.

Look at Flickr. It began as a multiplayer online game called The Game Neverending. Its creators soon realized the photo-sharing aspect of the game was a more plausible product than the game itself (which was eventually shelved). Be willing to admit mistakes and change course.

Be a surfer. Watch the ocean. Figure out where the big waves are breaking and adjust accordingly.

91: Start Your Engines

Done!

Alright, you made it! Hopefully you're psyched to start Getting Real with your app. There really has never been a better time to make great software with minimal resources. With the right idea, passion, time, and skill, the sky's the limit.

A few closing thoughts:

Execution

Everyone can read a book. Everyone can come up with an idea. Everyone has a cousin that's a web designer. Everyone can write a blog. Everyone can hire someone to hack together some code.

The difference between you and everyone else will be how well you execute. Success is all about great execution.

For software, that means doing a lot of things right. You can't just have good writing but then fail to deliver on the promises in your prose. Clean interface design won't cut it if your code is full of hacks. A great app is worthless if poor promotion means no one ever knows about it. To score big, you have to combine all these elements.

The key is balance. If you tilt too far in one direction, you're headed for failure. Constantly seek out your weak links and focus on them until they're up to par.

People

It's worth reemphasizing the one thing that we think is the most important ingredient when it comes to building a successful web app: the people involved. Mantras, epicenter design, less software, and all these other wonderful ideas won't really matter if you don't have the right people on board to implement them.

You need people who are passionate about what they do. People who care about their craft — and actually think of it as a craft. People who take pride in their work, regardless of the monetary reward involved. People who sweat the details even if 95% of folks don't know the difference. People who want to build something great and won't settle for less. People who need people. OK, not really that last one but we couldn't resist throwing a little Streisand into the mix.

Anyhow, when you find those people, hold onto them. In the end, the folks on your team will make or break your project — and your company.

More Than Just Software

It's also worth noting that the concept of Getting Real doesn't apply just to building a web app. Once you start grasping the ideas involved, you'll see them all over the place. Some examples:

- Special ops forces, like the Green Berets or Navy Seals, use small teams and rapid deployment to accomplish tasks that other units are too big or too slow to get done.
- The White Stripes embrace constraints by sticking to a simple formula: two people, streamlined songs, childlike drumming, keeping studio time to a minimum, etc.
- Apple's iPod differentiates itself from competitors by not offering features like a built-in fm radio or a voice recorder.
- Hurry up offenses in football pick up big chunks of yards by eliminating the "bureaucracy" of huddles and play-calling.
- Rachael Ray bases her successful cookbooks and TV show on the concept of 30-minute "Get Real Meals."
- Ernest Hemingway and Raymond Carver used simple, clear language yet still delivered maximum impact.
- Shakespeare reveled in the limitations of sonnets, fourteen-line lyric poems in iambic pentameter.
- And on and on...

Sure, Getting Real is about building great software. But there's no reason why it needs to stop there. Take these ideas and try applying them to different aspects of your life. You might just stumble upon some neat results.

Keep In Touch

Let us know how Getting Real works out for you. Send an email to gettingreal [at] basecamp [dot] com.

Also, stay up to date with the latest offerings from Basecamp by visiting Signal vs. Noise, our blog about usability, design, and a bunch of other stuff.

Thanks for reading and good luck!