How To Launch Software

August 22, 2008

Original link

37signals recommends that software developers pursue what they call the Hollywood Launch. They don't give any argument for this method, except perhaps the title (as if Hollywood was a business you should try to imitate?) — I guess the idea is that you're supposed to do it since 37signals says to.

The basic idea behind the Hollywood Launch is simple: you release a few hints about your product to build buzz, slowly revealing more and more until the big day, when you throw open the doors and people flood your site, sent there by all the blog coverage and email alerts.

This may work well for Hollywood — if your movie is a big hit at the box-office on opening weekend, then the movie theaters are more likely to keep showing it in the weeks to come and you get credit for being "one of the weekend's biggest films". But for software developers, it's moronic. Your software isn't being released in theaters, it's available over the Web. You don't have to worry about the theater no longer showing after week one; you can keep pushing it for years, growing your userbase.

Instead what happens when software developers try the Hollywood Launch, and I've seen this many times, is that users indeed do flood to your site on launch day but…

1. They bring the site down from the load. You scramble to get it back up and succeed by coding like a mad man, only to find…
2. They discover some big bug that you never quite noticed before, which makes the whole thing look like embarrassing hackwork. (*What? You forgot to test that last-minute JavaScript change in IE6 1/2?*) So you're desperately rushing to fix the bug before the traffic dies down, rush-patching things and restarting the server when…
3. You bring the site down for everyone because there was a syntax error in your patch that keeps the server from coming back up. You fix it while cursing yourself madly. Finally everything seems to work. You take a breath and decide to see what people are saying about you on the Web, only to discover…
4. Everyone misunderstood what your product does because your front page wasn't clear enough. Now they all think it's stupid and wonder aloud how you even know how to breathe. So you reply in all the comment threads and fix your front page to ensure no one could possibly misunderstand what it is you're doing just in time to find…

5. All the traffic is gone.

Tomorrow, hardly any of those users come back. Your traffic graphs look like the sharpest mountain you've ever seen: a huge climb up and then, almost immediately, a similarly-sized crash back down.

So what do you do then? Well, you do what you should have done all along: you grow the site.

I'll call this technique the Gmail Launch, since it's based on what Gmail did. Gmail is probably one of the biggest Web 2.0 success stories, so there's an argument in its favor right there. Here's how it works:

1. Have users from day one. Obviously at the very beginning it'll just be yourself and your co-workers, but as soon as you have something that you don't cringe while using, you give it to your friends and family. Keep improving it based on their feedback and once you have something that's tolerable, let them invite their friends to use it too.

2. Try to get lots of feedback from these new invitees, figuring out what doesn't make sense, what needs to be fixed, and what things don't work on their bizarre use case combination. Once these are all straightened out, and they're using it happily, you let them invite their friends. Repeat until things get big enough that you need to…

3. Automate the process, giving everyone some invite codes to share. By requiring codes, you protect against a premature slashdotting and force your users to think carefully about who actually would want to use it (getting them to do your marketing for you). Plus, you make everyone feel special for using your product. (You can also start (slowly!) sending invite codes to any email lists you might have.)

4. Iterate: give out invite codes, fix bugs, make sure things are stable. Stay in this phase until the number of users you're willing to invite is about the same as the number you expect will initially sign up if you make the site public. For Gmail, this was a long time, since a lot of people wanted invites. You can probably safely do it sooner.

5. Take off the invite code requirement, so that people can use the product just by visiting its front page. Soon enough, random people will come across it from Google or various blogs and become real users.

6. If all this works — if random people are actually happy with your product and you're ready to grow even larger — *then* you can start building buzz and getting press and blog attention. The best way to do this is to have some kind of news hook — some gimmick or controversial thing that everyone will want to talk about. (With reddit, the big thing was that we switched from Lisp to Python, which was discussed endlessly in the Lisp and Python communities and gave us our first big userbase.)

7. Start marketing. Once you start using up all the growth you can get by word-of-mouth (and this can take a while — Google is only getting to this stage *now*), you can start doing advertising and other marketing-type things to provide the next big boost in growth.

The result will be a graph that just keeps accelerating and climbing up. That's the graph that everyone loves to see: solid growth, not a one-day wonder. Good luck.

Since 37signals quotes from people who followed their advice, I thought I might as well do the same. mojombo:

> I find this to be excellent advice. This is exactly the approach we took at GitHub almost down to the letter. It took about 2 months until the site was good enough to use to host the GitHub source, another month until we started private beta with invites, and three more months until public launch.
>
> Artificial scarcity is a great technique to generate excitement for a product while also limiting growth to a rate that won't melt your servers. We worked through a huge number of problems and early users gave us some of the ideas that have defined GitHub. By doing a Hollywood launch, things would have been very different and I am convinced, very much worse.
>
> Do not, I repeat, DO NOT underestimate how much your users will help you to define your product. If you launch without having significant user feedback time, you've essentially thrown away a massive (and free) focus group study.
>
> Let me also say that when we finally did our public launch, there was plenty of buzz, and all of it was the RIGHT kind of buzz. The buzz that attracts real, lasting customers (and no, we weren't on TechCrunch, that traffic is garbage).