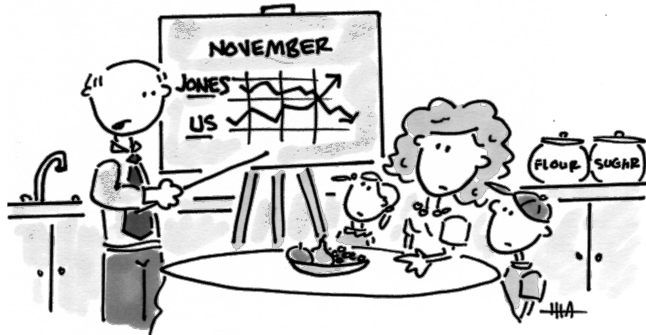


“ROI” is the wrong way to sell your product

by Jason Cohen on October 10, 2012

Customers ask for ROI calculations to justify purchasing your software, but it still doesn't convince them. Here's what to do instead.



"As you can see, we've not only kept up with the Jones, but surpassed them. Next up, the Nelsons."

I can't remember how many times at Smart Bear I tried to sell our main product Code Reviewer with the argument that it "saves you money." Some customers demanded it in the form of an ROI spreadsheet. So we gave it to them.

The argument makes sense (though it's wrong). Code Reviewer is tool which helps software developers review each other's work, just like an editor of a book. It cuts the time of a code review in half—maybe better—because it eliminates busywork. We even lead with that in our one-pager (because back in 2004, sales calls were in person, and one-page leave-behinds were useful):

CodeReviewer™

Cut peer code review time in half.

Typical Peer Review	CodeReviewer
Find a developer to perform the review. Either drag her into your office or package up all changes (both source code and diffs) and ship them to her. It's easy to miss a change. <div style="text-align: right; font-size: small;">10 minutes</div>	Pick the reviewer from a list. Changes are detected, packaged, and shipped automatically through version control integration. <div style="text-align: right; font-size: small;">1 minute</div>
She reviews the changes. If she's looking over your shoulder, you scribble down comments somewhere. If you've shipped the changes, she must record file/line numbers with comments. <div style="text-align: right; font-size: small;">20 minutes</div>	She reviews changes in unique before/after side-by-side view. She clicks on a line to leave a comment. <div style="text-align: right; font-size: small;">15 minutes</div>
You make corrections, manually locating the spot for each comment. If you want her to OK your corrections, you must repackage and ship everything. She can't tell what's corrected and what's the same as the last submission. <div style="text-align: right; font-size: small;">25 minutes</div>	You make corrections, jumping directly from her comments to your editor. Send your corrections back to her with a single click. She can view just the latest corrections if she prefers. <div style="text-align: right; font-size: small;">15 minutes</div>

Total: 45 minutes.
Problems: No audit trail. Insufficient tools for packaging and examining differences against whole source files. Hard to maintain and distribute a project checklist.

Save time every day with CodeReviewer

- ▶ Quickly package code for review
- ▶ Automate audit trails
- ▶ Enforce checklists
- ▶ Never miss a change again
- ▶ Review code remotely
- ▶ Integrates with version control system
- ▶ No server to manage

CodeReviewer automates the peer code review process so developers spend time looking at source code instead of managing the process itself. **CodeReviewer** is the only peer code review tool with support for multiple remote reviewers, automated audit trails, SCM integration, and custom reviewer checklists.

CodeReviewer is supported on Windows NT, 2000, XP, and 2003. It works with Perforce, SourceSafe, Subversion, CVS, and Vault version control systems.

Web: <http://code-reviewer.com>
E-Mail: info@code-reviewer.com
Phone: 877.501.5651

SmartBear Software

Copyright © 2004 Smart Bear, L.L.C. All Rights Reserved.
"SourceSafe" and "Windows" are trademarks of Microsoft Co. "Perforce" is a trademark of Perforce Software, Inc. "Vault" is a trademark of SourceGear L.L.C.

The economics are obvious: If two developers would each normally spend an average of 30 minutes on a code review, once a day, that's 20 hours per month. At a fully-loaded developer cost of \$150/hr, that costs you \$3,000/mo. Code Reviewer cuts the time in half, which means you save \$1,500 every month.

Code Reviewer costs \$499/developer—one time¹—so you make your money back with the first month! After that it's gravy... all the way to the bank.

¹ Ah, the days before SaaS, when all software was "lifetime," though you'd attempt "maintenance and upgrades" sales in future years at 20% of the original price. Life is better now.

You can't afford *not* to buy it, right?

But this argument never worked, not once, even though the reasoning is sound and customers requested it. The reason is that these “savings” aren’t how budgets actually work.

In a perfect world, if the software development organization really did produce more, higher-quality code, that’s undeniably valuable. But it’s also essentially impossible to measure, and is in fact not measured. So the “savings” are invisible, even if real.

On top of that, budgets are siloed; the “Salaries” budget is separate from the “Tools” budget. Never once in my seven years selling software at Smart Bear, whether 10 seats or 2500 seats, did a company make a decision in which savings in the “Salaries” budget resulted in an increase to the “Tools” budget.

With Code Reviewer, the customer buys because they genuinely believe it makes their team more productive, even though they cannot quantify ROI. Spreadsheets are worthless—they either believe, or they don’t.

As the startup founder, what do you do about this?

First, **create so much value² that there’s no need to “compute” it**. If so-called pay-back period really is less than a month, it’s obvious. Change someone’s workflow

so drastically for the better that they can’t live without it whether it’s saving money or not. Improve your customers’ marketing campaigns so obviously and drastically that they don’t need a spreadsheet to understand its value.

² more leads, more sales, faster code, efficiency, time-savings, cost-savings, delight, corporate brand, personal image, societal impact

Second, **price according to ability and willingness to pay and the structure of your business model, rather than as a direct function of so-called “value.”** The software development department has a budget for tools, and different companies have different ways of arguing internally for expanding that budget—you have to match those constraints. The marketing department might be willing to pay for services but not for tools.

In any case, rarely is a simple “cost savings” equation the way you’re going to win sales or set prices.

Printed from: *A Smart Bear*

<https://longform.asmartbear.com/roi-selling/>

© 2007-2024 Jason Cohen  @asmartbear