

# Show Me the Money

by Jeff Patton

SOMETIMES THE MOST NEGLECTED USERS of our system are the people who asked for it in the first place—our stakeholders. What happens when we forget that?

Say you're one of a dozen VPs of MegaCorp International. You've sponsored a software project to improve the time it takes customer service reps to take orders over the phone. Your marketing people told you that doing this would increase order sizes, improve your reputation, and, ultimately, add more customers. Your finance people told you it could earn the company several million dollars this coming year.

But things are looking rocky just now. The project delivered a few months late and a few hundred thousand dollars over budget. There are bugs—big bugs. The customer management people are moaning because they have to learn something new and they are already asking for changes and improvements. And to make matters worse, the executive team is hassling you to stop the hemorrhaging and shut the project down.

Then it hits you. You can make sure this thing is on the right track by making sure the time it takes to service customers did in fact decrease and that order sizes are increasing!

Slapping your forehead and muttering to yourself, "Why didn't I do this before?" you run down to the development department and ask, "How do I see the average time it takes to service customers and check the average size of orders placed through this new system?" Your lead developer responds: "Um . . . er . . . well, I'm not sure. I guess we could write a report. Well wait, we don't really keep track of that time-to-service-a-customer thing, and the system can't tell the difference between orders placed over the phone or via the online service. We'll have to



Jeff Patton says give stakeholders a way to keep score and they'll try to improve it.

change the software. But we could get it done in a month or so." Ouch.

I've seen this scene played out too many times. We set out with good intentions to build a piece of software to solve a business problem. We then focus on the people who'll be using the software to do their jobs and forget the reasons we built the software. When the time comes to prove the software is worth its weight in code, the software can't really tell us.

Suppose when you are building your software you consider your stakeholders as first-class users. Then suppose you build in functionality that allows your stakeholders to observe the very things that concern them.

As you're starting a new project or as you add to an existing one ask:

- How does this software earn our organization money?
- What can we measure and report to prove it?

- How do we track those measurements over time?
- How do we know when things are going poorly?
- How do we know when things are going well?

Design your software to gather and retain sufficient information to answer these questions—as it's busy earning you money. To keep stakeholders informed, design software with reports, status screens, executive dashboards, and email notification.

In addition, something magic happens when users can monitor their activities. If users are able to see information about their performance, they often work to improve it—speed up their own transaction times or increase their average order size. Give users a way to keep score and they'll try to improve it.

*(Continued on page 47)*

SEAN GRAFF

(Continued from page 48)

**SHOW ME THE MONEY**

Think of this as *adding gauges* to your software. Just like your car can tell you how fast it's going and how many miles it's been driven—your software should be able to tell you details about how it's being used.

There are a lot of examples that show this sort of thinking is becoming more commonplace. Google “executive dashboard” and you’ll find the companies that are already building software with one—and the companies that forgot them and are ready to add dashboards to the enterprise class products. These are gauges. These are stakeholder features. These features tell the software’s owners and users how well the software is meeting their goals.

If you’re responsible for designing and building software, don’t ignore the reasons you built the software in the first place. Ask your software to “Show Me the Money.” **{end}**

*Jeff Patton leads teams of Agile developers to build the best software possible. He proudly works at ThoughtWorks.*

**EDITOR’S NOTE:** *This issue’s “The Last Word” previews a column from our newest StickyMinds.com columnist, Jeff Patton. Jeff’s series of columns on software design and pre-design tips appears each quarter on the StickyMinds.com home page. Each week a new column from a different industry expert is featured on the site.*

# BETTER SOFTWARE

## Have the Last Word!

If you have a point to make or a side to take on issues and trends that affect the industry, we want to hear from you.

We are looking for insightful, thought-provoking commentary for possible use as a Last Word column.

Please send your submission to **editors@bettersoftware.com**. You will be notified if you are being considered for publication.

## Index to Advertisers

AutomatedQA .....	www.automatedqa.com .....	15
Agitar .....	www.agitar.com .....	39
Better Software Conference .....	www.sqe.com/bettersoftwareconf .....	9
Empirix .....	www.empirix.com .....	7
ICSQ .....	www.asq.org/softwareforum/conferences .....	45
Intel .....	www.intel.com .....	5
McCabe .....	www.mccabe.com/index_stqe.htm .....	Inside Front Cover
Mercury .....	www.mercury.com .....	Back Cover
Parasoft .....	www.parasoft.com/insure .....	11
PNSQC .....	www.pnsqc.org .....	44
Rally .....	www.rallydev.com/bsm .....	16
Seapine Software .....	www.seapine.com .....	2
Segue .....	www.segue.com .....	Inside Back Cover
Software Quality Engineering On-Site Training .....	www.sqe.com .....	43
Software Quality Engineering PowerPass .....	www.sqe.com .....	46
Software Quality Engineering Training .....	www.sqe.com .....	17
STAREAST2005 .....	www.sqe.com/stareast .....	37
Telelogic .....	www.telelogic.com .....	33

### Display Advertising

Shae Young syoung@sqe.com

### All Other Inquiries

info@bettersoftware.com

*Better Software* (USPS: 019-578, ISSN: 1532-3579) is published nine times per year. Subscription rate is US \$75 per year. A US \$35 shipping charge is incurred for all non-US addresses. See enclosed subscription cards for special offers. Payments made to Software Quality Engineering must be made in US funds drawn from a US bank. For more information, contact info@bettersoftware.com or call (800) 450-7854 or (678) 387-4645. Back issues may be purchased for \$15 per issue (plus shipping). Volume discounts available.

Entire contents © 2005 by Software Quality Engineering (330 Corporate Way, Suite 300, Orange Park, FL 32073), unless otherwise noted on specific articles. The opinions expressed within the articles and contents herein do not necessarily express those of the publisher (Software Quality Engineering). All rights reserved. No material in this publication may be reproduced in any form without permission. Reprints of individual articles available. Call for details.

Periodicals Postage paid in Orange Park, FL, and other mailing offices. POSTMASTER: Send address changes to *Better Software*, P.O. Box 923626, Norcross, GA 30010, info@bettersoftware.com.