



Managing a Software Business

Diomidis Spinellis

HUGE INDUSTRIES, from the automotive and healthcare industries to finance and entertainment, center increasingly on software. Managing such a software business, let alone the transition from an established one, is tough because software's ethereal nature offers infinite

options; a software–hardware combination (consider the iPhone); or a service, such as eBay? Having a service component in your offering lets you secure recurring revenue. However, you'll also need to invest in and manage the operations required to deliver it. Thanks to the Internet and

existing ones retrained, power structures and resources might be reallocated, and the organization's culture might need to adjust to the new way of doing things. Most dramatically, a software-centered business model might not be able to deliver the comforts of an existing business model. It might involve painful redundancies and adjustments for shrunk revenue and profit margins.

Then, consider your market. You can target enterprises or individuals, you can create a specialized product or appeal to the masses, you can cover a broad area or focus on a niche, and you can lead the market or complement an existing offering. Two features of software can guide you here. The low marginal distribution cost (the cost of adding another user) means that additional sales can often directly translate into pure profit. However, entering a mass market is hard; for example, more than two-thirds of the millions of mobile phone apps are never or hardly ever downloaded. Software is also easy to distribute on a global scale. This can allow you to focus on a profitable niche.

You need to decide who builds your offering, by drawing your organization's boundaries.

lucrative or catastrophic choices. A map of software business management tasks can help you navigate the securest and most profitable route through the sea of alternatives.

Your Business Model and Strategy

Start with your business model, the idea that describes how you create value. Will that be a pure software product, such as Microsoft Win-

cloud computing, it's now easy to include a service component into any type of software.

If your business model aims to supplant an existing one (think of Netflix moving from posted DVDs to video streaming), you must also plan and manage the transition and its upheaval. Existing revenue streams might dry up before new ones start bringing in cash, new people might be brought in and ex-

EDITORIAL STAFF

Lead Editor: Brian Brannon,
bbrannon@computer.org

Content Editor: Dennis Taylor

Staff Editors: Lee Garber, Meghan O'Dell,
and Rebecca Torres

Publications Coordinator:
software@computer.org

Lead Designer: Jennie Zhu-Mai

Production Editor: Monette Velasco

Webmaster: Brandi Ortega

Multimedia Editor: Erica Hardison

Illustrators: Annie Jiu, Robert Stack,
and Alex Torres

Cover Artist: Peter Bollinger

Director, Products & Services:
Evan Butterfield

Senior Manager, Editorial Services:
Robin Baldwin

**Manager, Editorial Services Content
Development:** Richard Park

Senior Business Development Manager:
Sandra Brown

Senior Advertising Coordinators:
Marian Anderson, manderson@computer.org
Debbie Sims, dsims@computer.org

CS PUBLICATIONS BOARD

David S. Ebert (VP for Publications), Alain April,
Alfredo Benso, Laxmi Bhuyan, Greg Byrd,
Robert Dupuis, Jean-Luc Gaudiot, Ming C. Lin,
Linda I. Shafer, Forrest Shull, H.J. Siegel

MAGAZINE OPERATIONS COMMITTEE

Forrest Shull (chair), M. Brian Blake, Maria Ebling, Lieven Eeckhout, Miguel Encarnação, Nathan Ensmenger, Sumi Helal, San Murugesan, Yong Rui, Ahmad-Reza Sadeghi, Diomidis Spinellis, George K. Thiruvathukal, Mazin Yousif, Daniel Zeng

Editorial: All submissions are subject to editing for clarity, style, and space. Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *IEEE Software* does not necessarily constitute endorsement by IEEE or the IEEE Computer Society.

To Submit: Access the IEEE Computer Society's Web-based system, ScholarOne, at <http://mc.manuscriptcentral.com/sw-cs>. Be sure to select the right manuscript type when submitting. Articles must be original and not exceed 4,700 words including figures and tables, which count for 200 words each.

IEEE prohibits discrimination, harassment and bullying: For more information, visit www.ieee.org/web/aboutus/whatis/policies/p9-26.html.



WELCOME NEW ED BOARD MEMBER



Nicholas A. Kraft is a principal scientist in the Industrial Software Engineering group at ABB Corporate Research. Previously, he was an associate professor in the University of Alabama's Department of Computer Science. In these roles he has led research projects for ABB, DARPA, and the US National Science Foundation. His research interests include software maintenance and evolution, empirical software engineering, and end-user software engineering. Kraft received a PhD in computer science from Clemson University. He's a senior member of ACM and IEEE. He will be responsible for setting up and managing *IEEE Software's* Journal First, Conference Second (J1C2) process, whereby top *Software* articles will be presented to conferences after their publication. Contact him at nicholas.a.kraft@us.abb.com.

A key part of any strategy is how you deal with competition. Software, and especially a service built through it, exhibits strong network effects: your offering's value increases with the number of its users. Positioning yourself as a hub in a winner-takes-all field can be remarkably rewarding. A related strategy involves offering a platform or an ecosystem in which your business can grow together with others that depend on yours. Technology can play an important role here. The choice of open or proprietary file formats, APIs, and plug-in mechanisms can make the difference between failure and success.

Finally, as part of your strategy, you need to decide who builds your offering, by drawing your organization's boundaries. Software construction might appear easy to outsource, but then what value are you offering? Also, given the big productivity differences among developers, your engineers will likely be a strategic

asset rather than just a cog in your operations' machinery. You might be able to minimize licensing costs by using or modifying open source components, but so can your competition. You can even crowdsource software development and support, but building and maintaining a live volunteer community isn't for the fainthearted.

Your Offering

You implement your strategy by managing the product or service you're offering. This is a delicate balancing act. An important guide is obviously the set of features your existing or potential customers have requested. Less obviously, many other things determine your offering's quality: its usability, security, maintainability, reliability, and efficiency. All these elements are at odds with each other, and all drive up costs. Piling features generally reduces the software's other qualities. Also, increasing security reduces usability, increasing efficiency

IEEE SOFTWARE WINS APEX 2016 AWARD OF EXCELLENCE



IEEE Software has received an APEX 2016 Award of Excellence in the “Magazines, Journals, & Tabloids—Electronic” category. APEX 2016—the 28th Annual Awards for Publication Excellence—is an international competition that recognizes outstanding publications from newsletters and magazines to annual reports, brochures, and websites.

According to the APEX 2016 judges, “The awards were based on excellence in graphic design, quality of editorial content and the success of the entry in conveying the message and achieving overall communications effectiveness.”

The award went to our November/December 2015 issue, which focused on refactoring, the process of improving a program’s source code without changing its external behavior. Contributors to this special issue included William Griswold and William Opdyke, whose separate PhD dissertations contained the initial research that led to refactoring becoming a standard practice.

The issue looked at how far refactoring has come since then and addressed industry perspectives on the adoption of refactoring, refactoring for software on mobile devices, and lessons from the trenches on database refactoring. The “Point/Counterpoint” column provided two opposing viewpoints: one from John Brant that “Refactoring Tools Are Trustworthy Enough” and one from Friedrich Steimann that “Trust Must Be Earned.”

agile process, whereas if they value certainty and stability, you might want to go for a more predictive process. Your organization’s culture and staff’s experience will play a role here. Your job also includes adapting the process to its environment, assessing its effectiveness and efficiency, and continuously improving it.

On the basis of the process you’ve chosen, you need to manage the corresponding planning and enactment. Predictive processes will require you to invest time in process planning, deliverable definitions, effort and schedule estimations, and resource allocation, as well as the management, measurement, and control of risks, quality, and your plan. An agile process will be even more demanding. Here, you must bring the customer close to the developers, ensure the software’s frequent delivery, motivate individuals, empower teams to organize themselves, and promote a culture of technical excellence.

With software increasingly being offered as a service, another important thing you must manage is the operations behind the delivery. Here, you need to coordinate people and technology to ensure the reliable delivery of a high-quality service. Elements to keep an eye on include infrastructure configuration management, deployments, security methods to address the inevitable failures, the user experience and application performance monitoring, issue ticketing, and around-the-clock support.

Your processes are ultimately run by people, and managing them in a software business is no mean feat. As clever specialized knowledge workers, developers can’t be effectively managed through command and control. The tools you’ll

reduces maintainability and reliability—the list goes on. It’s your job as a manager to make sensible choices, triage, prioritize, and handle expectations.

If this wasn’t difficult enough, you also need to consider your offering’s stability and evolution. Users often distrust a stale product, whereas enterprises typically value stability (until they ditch you for the new kid on the block). Swift technology advances and Lehman’s laws of software evolution guarantee that, unless you constantly invest in your software, it will quickly become ob-

solete. Cruelly, each year your competitors will find it easier to build a competing product, while your software becomes less flexible and attractive. At some point, the frightening possibility of starting afresh might be the only viable option.

Your Processes

To deliver your software or service, you must manage its development and operations. Start by choosing the software development method you’ll follow. A lot depends on your customers. If they seek adaptation to changing requirements, select an



THE *IEEE* SOFTWARE BLOG

If you haven't been reading the *IEEE Software* Blog (<http://blog.ieeesoftware.org>) over the last two months, here's what you've missed:

- "Diversity in CS Shines at NSBC 2016"
- "CodeTube: Making the Best of Software Development Video Tutorials"
- "There Ain't No Such Thing as a Free Build"
- "Insights for Practitioners into Empirical Software Architecture Research"
- "Has Empiricism in Software Engineering Amounted to Anything?"
- "Does Your Android App Collect More Than It Promises To?"

Let the blog help you keep up with the quickly changing world of software engineering.

use are motivation, empowerment, and trust. Keeping developers productive should be your number-one priority. Eliminate distractions and waste, give them all the tools and hardware they need, and set up your organization's structures in a way that lets them write great software. Avoid promoting stellar developers to become mediocre managers, by having a separate engineering career track. Attracting talent in a highly competitive marketplace, sharing your organization's values among all developers, and keeping the expensively acquired and trained devel-

opers in your organization are also critical tasks.

Managing a software business is uniquely challenging and rewarding because you work with people to sprinkle our world with software pixie dust. Make it so! 🍷



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.

We welcome your letters.

Send them to software@computer.org. Include your full name, title, affiliation, and email address. Letters are edited for clarity and space.



HOW TO REACH US

AUTHORS

For detailed information on submitting articles, write for our editorial guidelines (software@computer.org) or access www.computer.org/software/author.htm.

LETTERS TO THE EDITOR

Send letters to

Editor, *IEEE Software*
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
software@computer.org

Please provide an email address or daytime phone number with your letter.

ON THE WEB

www.computer.org/software

SUBSCRIBE

www.computer.org/software/subscribe

SUBSCRIPTION CHANGE OF ADDRESS

Send change-of-address requests for magazine subscriptions to address.change@ieee.org. Be sure to specify *IEEE Software*.

MEMBERSHIP CHANGE OF ADDRESS

Send change-of-address requests for IEEE and Computer Society membership to member.services@ieee.org.

MISSING OR DAMAGED COPIES

If you are missing an issue or you received a damaged copy, contact help@computer.org.

REPRINTS OF ARTICLES

For price information or to order reprints, email software@computer.org or fax +1 714 821 4010.

REPRINT PERMISSION

To obtain permission to reprint an article, contact the Intellectual Property Rights Office at copyrights@ieee.org.