Editor in Chief: **Diomidis Spinellis**
Athens University of Economics and Business, dds@computer.org

The Strategic Importance of Release Engineering

Diomidis Spinellis

RELEASE ENGINEERS are building pipelines to deliver software products from the developers to the end users. This is the apt metaphor that John O'Duinn used to define release engineering. O'Duinn, who was Mozilla's director of release engineering for more than six years, was delivering a keynote on release engineering's value as a force multiplier at the 2014 USENIX Release Engineering Summit. He nailed the point.

Some used to consider release engineering a mundane activity, one that appealed to control freaks wishing to play God with other developers. How times change! In our era, where all nontrivial devices are controlled by software, more than a billion consumers can buy apps with a touch of a button, and software is delivered or served instantaneously through the Internet, release engineering has become strategically important. It affects the software we build, how we build it, and how we can make money out of it.

> A timely introduction of a clunker and a delayed entry of a masterpiece can destroy a product's chances of success.

Making Software ...

First consider agility. Reaping its many benefits requires smoothly running release-engineering machinery. Significant practices of agile software development include the coevolution of requirements and solutions, early delivery, and rapid, flexible response to change. These practices depend on the effortless creation of frequent software releases. Release plans that follow the "release early, release often" principle can increase a team's velocity, resulting in a beneficially tight feedback loop between developers and users. In some cases, concurrent, real-time, alpha-beta testing of new product features across slightly different releases can become a powerful oracle to guide product evolution.

Then think about software quality. The adroit introduction of features in new releases can improve the software's functionality, usability, and efficiency. In contrast, a bungled release can spell disaster. Dexterous software stabilization and issue triaging as part of release engineering will balance the new features with the required level of software reliability, maintainability, and portability. (Portability is again becoming important in cross-platform development markets, such as those for games and apps.)

Finally, take into account developer productivity. Inept release engineering can frustrate developers with protracted code freezes, agonizingly slow and brittle builds, and tiny infrequent time windows in which to commit their code to a production branch. This state of affairs is like pouring tar on developers' keyboards. Worse, besides holding back progress, it saps staff morale and increases turnover. Many sickly software shops have open release-engineering wounds.

... And Making Money

For highly innovative products and those facing cutthroat competition, time to market can make or break a launch. With potential customers globally connected through social networks like never before, firms rarely get a second chance to correct botched moves. Both a timely introduction of a clunker and a delayed entry of a masterpiece can destroy a product's chances of success. Adept release engineering can balance time pressures

0740-7459/15/$31.00 © 2015 IEEE

WELCOME NEW BOARD MEMBER AND DEPARTMENT EDITORS

I'm honored to welcome Marian Petre to the Editorial Board as our new associate editor in chief for human factors and also to congratulate Rafael Prikladnicki on his transfer from the Advisory Board to the Editorial Board as the department editor for the Voice of Evidence column. In addition, I'm excited to have Cesare Pautasso and Olaf Zimmermann serving as coeditors of the Insights column.

Marian Petre is a professor of computing in the Faculty of Mathematics and Computing at the Open University. She is also a past director of its Centre for Research in Computing. She has made contributions not only to software engineering research but also to software education and skill development for young researchers. She has also served as a guest editor for an *IEEE Software* special issue.

Rafael Prikladnicki is an associate professor at the Pontifica Universidade Catolica do Rio Grande do Sul and the director of TECNOPUC, the university's science and technology park. In this position he leads work at the intersection of academia and commercial software development. He is also leader and cofounder of the MuNDDoS Research Group on Distributed Software Development.

Cesare Pautasso is an associate professor at the Faculty of Informatics at the University of Lugano. His research group focuses on building experimental systems to explore the architecture, design, and engineering of next-generation Web information systems. His teaching, training, and consulting activities cover advanced topics related to emerging Web technologies, RESTful business process management, and cloud computing. Pautasso is a senior IEEE member and an advisory board member of EnterpriseWeb.

Olaf Zimmermann is a professor and institute partner at the Institute for Software at the University of Applied Sciences (HSR FHO) in Rapperswil. His areas of interest include Web-based application and integration architectures, SOA and cloud design, and architectural knowledge management. He is an author of *Perspectives on Web Services* (Springer, 2003) and contributed to several IBM Redbooks, including the first one on Eclipse and Web services (2001).

Please join me in congratulating Marian, Rafael, Cesare, and Olaf in their new positions.

with demanding software quality requirements, ensuring that the right product is launched at the right time.

Then comes marketing. Its executives who cover product packaging with sparkly "New!," "Improved!," and "Better Tasting!" labels are surely onto something. In demanding marketplaces, products that don't move forward sufficiently fast will stall and crash. Release engineering can assist marketing though the regular, reliable delivery of new features from a product's evolving roadmap. If you think this task is easy, consider that at the same time, current and legacy versions must also be serviced through bug fixes and urgent security patches. When done right, release engineering promotes openness by replacing vacuous "The Greatest Ever!" claims with concrete, accurate lists of features and fixes (and gotchas) associated with each new product version.

Crucially, release engineering can also form the basis of an entire business model. Some companies derive most of their revenue not from initial purchases but from product updates or subscriptions. In these cases, release engineering isn't supporting the product, release engineering *is* the

product. An important niche in this area is updates for software assets other than code, which must also be put under release engineering's control: virus definitions, football game player data packs, playlists, fonts, clipart, and so on. Business models based on frequent updates keep customers engaged and loyal, reduce the risk from shifting markets, and, by rapidly delivering features, allow for customer-driven innovation.

Finally, release engineering affects client relations and the company and product image. A slipshod release can lead to data loss and service disruption among customers or, in extreme cases, transform working appliances into useless bricks. On the other hand, a lack of visible progress in a software product can taint it as "abandonware." Few companies can successfully deal with the fallout and escape the damage caused by the consequent reputation damage. Timely, reliable, new releases keep happy those customers who live on the leading edge, while painless fixes and backward compatibility maintain a warm fuzzy feeling with the more conservative group.

Challenges

If you think the demands on release engineering are tough, wait till you see some additional challenges. In some application domains, regulators can prescribe software requirements, time schedules, and a lengthy, cumbersome approval process for new software releases. So-called 0-day security threats exploit vulnerable applications the same day the vulnerability becomes known, pressuring release engineers for a lightning-fast reaction. Then, there are users who, rightly, hate the disruption of frequent updates and want to be kept forever on long-term support branches (properly maintained, of course).

Add to this mix a rapidly evolving ecosystem of hardware, standards, operating systems, APIs, libraries, databases, and middleware, with its own unique pressing demands for maintaining compatibility. Supporting simpler embedded devices isn't easier because these might offer limited or intermittent connectivity and are often tended by untrained users—a ticking-time-bomb combination. Finally, within their organization, release engineers must support legacy tools and even hardware that might be required to build and maintain older versions.

Coming out with methods to address these challenges will probably require a lot more work and another *IEEE Software* theme issue in a few years. However, one thing is certain: release engineering is anything but boring; indeed, it can be an organization's hidden strategic asset.

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

HOW TO REACH US

WRITERS

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.