

continued from p. 96

Encouraging Participation

A key failing of management today is the belief that teams can organize their own affairs at the team level, but their input in the managerial decisions at higher—project or organizational—levels isn't required. As such, although many managers are happy to trust their teams with technological decisions, such as picking the best programming platform or hardware, they're hesitant to involve them in managerial decisions such as negotiating project contracts.

What does a software development team have to do with how their managers negotiate business contracts? A common practice in contract negotiation has been for senior management—through the sales teams—to negotiate fixed-bid contracts for IT projects without upfront team consultations. A consequence is that teams become severely restricted in their ability to respond to inevitable changes in customer requirements during the project and fail to manage scope changes effectively, which ultimately increases costs and drives down profit margins.

Software teams usually find it “demotivating to be given ridiculous deadlines” by managers who “don't actually have a clue about the techni-

cal challenges associated with them.”¹ Fixing this requires management to be aware of such sentiments among the software practitioners they manage and to encourage participation in providing estimates for projects based on their technical expertise and experience. After all, your team actually implements the project—who better to provide estimates of the time and effort involved?

Sponsoring Innovation

Teams look up to senior management for the project's overall guiding vision, but they're capable of the implementation themselves. After all, if you've hired the top talent for your company, why undermine their ability to come up with truly innovative solutions by giving them templates? A common sentiment echoed by software developers today is that “we aren't code monkeys... tell us the what, not the how!”

Facing new challenges and designing innovative solutions is what makes work interesting for self-organizing teams and drives them to excellence. Coming up with business solutions and details of the technical implementation fosters a strong sense of ownership and responsibility among team members. Teams are much more likely to take ownership of something they came up

with than a solution that was forced on them by management.

In this age of social media, tools such as Facebook, Twitter, and Google+ give their users opportunities to network without boundaries, express opinions, make an impact, and give and receive feedback with global reach. However, this empowerment might be in direct conflict with a more conservative, hierarchical, and old-fashioned organizational culture, giving rise to discontent. A revolution in this sense could comprise of a loss of the best talent, which will often gladly move on to freer, more open environments. ☞

Reference

R. Hoda et al., “Self-Organizing Roles on Agile Software Development Teams,” *IEEE Trans. Software Eng.*, //VOL? ISSUE? PAGES?//, 2012.

RASHINA HODA leads the Software Engineering Processes Tools and Applications (SEPTA) research group at the University of Auckland. Her research interests include self-organizing teams, agile software development, and software project management. Hoda received her PhD in computer science from Victoria University of Wellington, New Zealand. Contact her at r.hoda@auckland.ac.nz.



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

IEEE Software (ISSN 0740-7459) is published bimonthly by the IEEE Computer Society. IEEE headquarters: Three Park Ave., 17th Floor, New York, NY 10016-5997. IEEE Computer Society Publications Office: 10662 Los Vaqueros Cir., Los Alamitos, CA 90720; +1 714 821 8380; fax +1 714 821 4010. IEEE Computer Society headquarters: 2001 L St., Ste. 700, Washington, DC 20036. Subscription rates: IEEE Computer Society members get the lowest rate of US\$56 per year, which includes printed issues plus online access to all issues published since 1984. Go to www.computer.org/subscribe to order and for more information on other subscription prices. Back issues: \$20 for members, \$209.17 for nonmembers (plus shipping and handling).

Postmaster: Send undelivered copies and address changes to *IEEE Software*, Membership Processing Dept., IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854-4141. Periodicals Postage Paid at New York, NY, and at additional mailing offices. Canadian GST #125634188. Canada Post Publications Mail Agreement Number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8, Canada. Printed in the USA.

Reuse Rights and Reprint Permissions: Educational or personal use of this material is permitted without fee, provided such use: 1) is not made for profit; 2) includes this notice and a full citation to the original work on the first page of

the copy; and 3) does not imply IEEE endorsement of any third-party products or services. Authors and their companies are permitted to post the accepted version of IEEE-copyrighted material on their own web servers without permission, provided that the IEEE copyright notice and a full citation to the original work appear on the first screen of the posted copy. An accepted manuscript is a version which has been revised by the author to incorporate review suggestions, but not the published version with copyediting, proofreading, and formatting added by IEEE. For more information, please go to: http://www.ieee.org/publications_standards/publications/rights/paperversionpolicy.html. Permission to reprint/republish this material for commercial, advertising, or promotional purposes or for creating new collective works for resale or redistribution must be obtained from IEEE by writing to the IEEE Intellectual Property Rights Office, 445 Hoes Lane, Piscataway, NJ 08854-4141 or pubs-permissions@ieee.org. Copyright © 2013 IEEE. All rights reserved.

Abstracting and Library Use: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use of patrons, provided the per-copy fee indicated in the code at the bottom of the first page is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.



Power to the People

Rashina Hoda



THE PERIOD BETWEEN 2011 and 2012 emphatically showcased the power of the people. From political upheavals in the Middle East to protests against the economic crisis in Europe to the Occupy Wall Street movement in the US to the anticorruption protests across India, the world witnessed several revolutions against multiple aspects of misgovernment and the undeniable desire of people to take matters into their own hands.

Does management in the software industry await a similar future? We can translate two major lessons learned from the upheavals in 2011 and 2012 to the software development world: the burgeoning desire of teams to participate in the management of their own affairs and the inability of managers to adapt their style in this new age. The fundamental problem is a widening gap between these two groups.

The first step to avoiding (or reconciling) this gap in the software development world is for senior management to acknowledge that teams no longer live—or want to live—in a world of command and control. More importantly, they want to be self-organizing and have adaptive, supportive, and collaborative leadership guiding them.

Building a Culture of Trust

With the rise of agile practices, self-organizing teams are fast becoming the way in which software teams want to work. Such teams are characterized by self-assignment of tasks and collaborative estimation, planning, and decision making; they also display a greater sense of ownership and responsibility. But self-organizing teams can't be leaderless or uncontrolled—instead, they require

a hands-free yet still watchful management style. Embracing this new age of management requires a balancing act between what “hands free” should yield and what “watchfulness” must monitor.

Self-organizing teams demand informal organizational structures, where the boundaries of hierarchy don't prohibit the free flow of information and feedback. In other words, while it's fine to have organizational titles such as CEO, product manager, or architect, the organizational structure itself should be informal in practice. At a minimum, senior management should make itself directly accessible to all employees (maintaining an open-door policy) and accept both positive and negative feedback.

Simply put, starting with an informal structure has a cascading effect. It leads to openness (marked through constant communication and feedback), which in turn leads to a culture of trust. In such organizations, senior management gives workers the freedom to voice opinions, raise concerns, seek support in resolving those concerns, and make collaborative decisions. Several mechanisms enable openness, such as user groups and mailing lists within the organization to encourage everyone to share and discuss experiences, challenges, and solution strategies. Other options include holding regular, one-on-one meetings to gain an understanding of the primary concerns and giving regular feedback on performance (instead of annual “gotcha” appraisals with little opportunity for self-improvement).

continued on p. 95