

software. The ecosystem received an evolutionary supercharge in the late 1990s as the generative PC became a gateway to the generative Internet.

By refusing to limit themselves to specific purposes and by welcoming contributions from disparate sources, the PC trounced stand-alone word processors like the Friden Flexowriter; the Internet trounced proprietary networks like CompuServe, MCI Mail, and Prodigy; and general-purpose online markets and gathering places overwhelmed their niche-specific counterparts. (Remember when Amazon.com sold only books?)

Unfortunately, the uncertainty fueling this proliferation of software and services is fading fast, making the IT industry less innovative and diverse. There are three reasons.

First, many players now believe they've mastered the fundamental uses of the Internet and personal computing. Confident they know what will win and what won't, they try to become gatekeepers for successful products rather than platforms for all comers. Producing a commodity OS isn't enough for Microsoft and Apple; they want to dominate the market for applications like Office and iTunes and beat out, subsume, or license third-party efforts for popular software. Many emerging video game, cell phone, and PDA platforms are closed from the start—third-party developers either aren't welcome or are subject to stringent licensing requirements.

Similarly, Internet infrastructure providers don't want to stop at Internet service. As the chairman of IDT put it in January of 2002, "Sure, I want to be the biggest telecom company in the world, but it's just a commodity. I want to be able to form opinion. By controlling the pipe, you can eventually get control of the content." That control means picking what data will flow and what won't, which in turn limits the ability of a wizard in a computer lab somewhere to invent an application that takes the world by storm.

Second, security threats have become genuinely overwhelming. The openness that enabled innovation has led to unacceptable vulnerabilities as consumer PCs have gained processing power and always-on high-bandwidth Net connections. A user clicking on the wrong .exe can entirely compromise his or her computer—transforming it into a networked zombie spewing spam, viruses, or denial-of-service attacks against other network targets.

Finally, the Internet and PCs attached to it threaten creative destruction to settled interests. Intellectual property owners, for example, don't want to see their works pirated through innovations like peer-to-peer software. And the publishers and lawmakers they then enlist to constrain such technologies care little for the

collateral damage done to the work of citizen journalists and bloggers, as well as other benefits that flow from P2P.

These forces benefit from limiting the flexibility of generic platforms. Thus, Internet service providers are asked by institutional copyright holders to terminate access to users suspected of infringing copyright or to prevent certain types of network traffic entirely. OS manufacturers create "trusted" platforms that can handle intellectual property with a minimum of leakage. And as security concerns mount, IT companies seek to save users from themselves by designing roadblocks that won't let PCs run just any program or handle just any data.

What ought to be done? Openness proponents must address security concerns,

TEAMS

Trust, but Verify

Trust among team members is good—usually. But with some teams, too much trust can actually depress performance, finds Claus Langfred, a professor of organizational behavior at Washington University in St. Louis.

Langfred surveyed 71 self-managing teams of MBA students to measure levels of trust, self-monitoring, and autonomy within them. The teams worked for four months on financial analyses, marketing projects, business case write-ups, and other projects and at the end competed in presenting them to faculty and industry experts. As self-managing teams, they had complete discretion in deciding how to carry out assignments. Langfred found that trust dampened performance most in teams whose members were highly autonomous—that is, those whose members worked independent of one another. Not surprisingly, he found that when these team members trusted each other, they tended not to monitor one another much. As a result, they had relatively low awareness of each other's activities, which affected performance, probably by hampering processes and coordination.

This suggests that in a specific type of team—one where members are both highly independent and trusting of one another—deliberate monitoring is important. Even if members of such teams do suspect that supervision would be wise, they may be uncomfortable suggesting it. Failing to keep an eye on team members' activities can be naive, Langfred concludes, regardless of levels of trust. So managers may want to require a modicum of oversight rather than let a team decide for itself. A little skepticism never hurt anyone—or any team.

—Gardiner Morse
Reprint F0505B

Harvard Business Review and Harvard Business School Publishing content on EBSCOhost is licensed for the individual use of authorized EBSCOhost patrons at this institution and is not intended for use as assigned course material. Harvard Business School Publishing is pleased to grant permission to make this work available through "electronic reserves" or other means of digital access or transmission to students enrolled in a course. For rates and authorization regarding such course usage, contact permissions@hbsp.harvard.edu