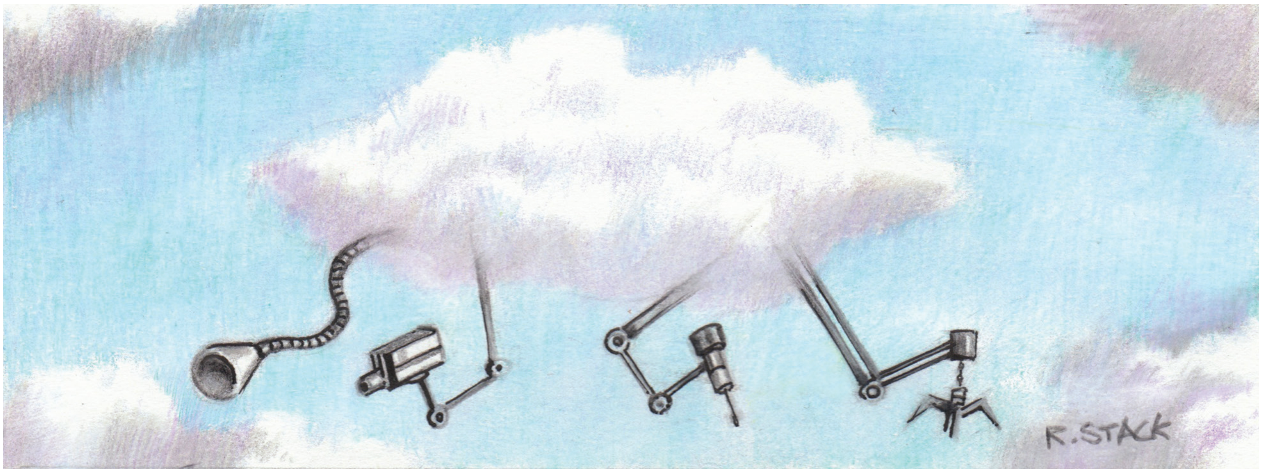
Editor: Diomidis Spinellis
Athens University of Economics
and Business. dds@aueb.gr



Developing in the Cloud

Diomidis Spinellis

RUNNING A TOP-NOTCH software development organization used to be a capital-intensive endeavor, requiring significant technical and organizational resources, all managed through layers of bureaucracy. Not anymore. First, many of the pricey systems and tools that we developers need to work effectively are usually available for free as open source software. More importantly, cheap, cloud-based offerings do away with the setup, maintenance, and user support costs and complexity associated with running these systems.

Post your comments online
by visiting the column's blog:
www.spinellis.gr/tools

Here are just a few of the services and providers that any developer group can easily tap into (you can find many more listed at http://goo.gl/slg2YY):

- version control (Bitbucket and GitHub);
- issue tracking, project management, and collaboration (Asana, Basecamp, FogBugz, GitHub, JIRA, Pivotal Tracker, Redmine, Trello, and YouTrack);
- remote application monitoring (BugSense, Crittercism, Exceptional, New Relic, and Sentry);
- localization (Pootle, Transifex, and WebTranslateIt);
- real-time communication (Google Hangouts, Grove.io, HipChat, Skype, and sqwiggle);
- email (Gmail, Outlook.com, and Yahoo! Mail);
- wiki (Confluence, Bitbucket, and GitHub);
- collaborative editing (Etherpad, Google Docs, Stypi, and Microsoft Office 365);
- file sharing (Box, DropBox, Google Drive, and SkyDrive);
- deployment servers (Amazon Web Services, Google App Engine, Heroku, Microsoft Azure, Nodejitsu, and Rackspace);
- store fronts (Google Play Store, iOS/Mac App Store, and Windows Store);
- payment processing (Braintree, Chargify, PayPal, and Stripe);
- IT support (Desk.com, Helpscout, and Zendesk);
- bulk email delivery (Campaign-Monitor, Mailchimp, mailgun, and Sendgridbulk);
- A/B testing (Optimizely);
- programming (Cloud9 and Visual Studio Cloud); and
- continuous integration (Bamboo, Cloudbees, Codeship, and Travis).

In addition, by hosting development servers on the cloud or by adopting the Vagrant virtual development environment configurator, a modern software shop can do away

0740-7459/14/$31.00 © 2014 IEEE

with the headache of setting up programmer workstations. Developers can simply use their own favorite device to connect to the team's cloud-based or virtualized preconfigured development setup.

The Meaning of Clouds

Life in the cloud has risks and costs different from those you face on earth. However, dealing with them is rarely a big deal, especially for smaller organizations, which often face greater risks by juggling resources to provide similar services in-house. The most important issues concern control of the data you store and the services you use. In short, don't bet your farm on a single obscure service provider. Select popular providers that support open data formats and that make your data available through standard protocols and APIs. This allows you to move elsewhere with minimal disruption if the service faces severe problems or if the cloud-based provider dissolves into thin air, as it were. Recognize vendor lock-in strategies and plan around them. Then you can have the peace of mind to enjoy the cloud's many benefits.

Cloud-based tools dramatically lower the capital requirements and setup costs of a software development organization. Therefore, if you have a neat idea for a software-based product or service, you can put together the development infrastructure in days (if not hours), armed with just a credit card. Lower barriers to entry mean that, with the right combination of skills, it's dramatically easier to develop your idea and bring it to market today than it has ever been.

Cloud-based services also help you employ best practices in each domain simply by registering as a user of a corresponding service. This gives you mature and polished tools and processes from the start. It's therefore inexcusable nowadays to track bugs through Post-it notes, to store revisions by copying files, or to collaborate on a document through email exchanges.

You can put together the development infrastructure in days (if not hours), armed with just a credit card.

Clouds and Organizations

If you work in a large organization, cloud-based development can affect you in two ways. First, your group might be able to benefit from these tools. In theory, a large organization has the budget, staff, and hardware to provide in-house alternatives to the cloud-based tools I outlined earlier. In practice, however, conflicting priorities, allocation constraints, conservative managers, sunk costs, and departmental intransigence might force you to use makeovers of green-terminal applications or 1990s client-server technology. Instead of fighting with the IT department to deploy the software tools you need, you might be able to purchase them as a cloud-based service using your department's budget or even petty cash. (Getting the required IT policy approvals and firewall configurations can be trickier.) You can also use the existence of these services as a credible threat to bargain for the provision of better in-house support. A second way cloud-based development can affect your organization is by drawing away talent. Your most creative developers no longer feel that your company offers them an exclusive cocoon where all their development needs are taken care of. On the contrary, they may suspect that by venturing out on their own they would get a more productive working environment through cloud-based services.

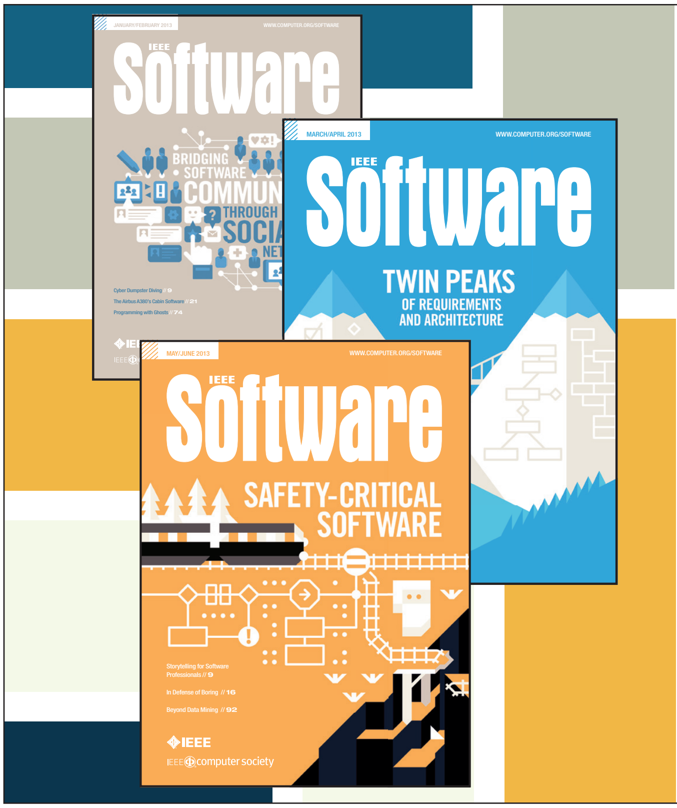
Using a cloud-based service also means fewer worries regarding scalability. Many of the services I named have hundreds of thousands of users. If your team grows from three people in a garage to 103 people in a hangar, cloud-based services can easily continue to accommodate you. Likewise, you don't have to settle for a cheap, second-rate, small-scale tool in the beginning to avoid the cost of the larger one: you can play in the major league from day one. The same goes for serving your customers from the cloud. When you provision your own infrastructure, you have to plan ahead for your hardware purchases and software licenses for the foreseeable growth in terms of volume of traffic and processing. Thus, by definition, your infrastructure will be underutilized in the beginning (because, remember, you planned ahead) and risks running out of steam too early if growth is stronger than what you anticipated. In the cloud, you can provision exactly the resources you need at each point and ramp up capacity in hours with a click of a button. The large market served by cloud-based services absorbs most of your required elasticity's cost.

From your customers' perspective, delivering a service through the cloud allows you to have a much closer relationship with them. You can issue new releases and fix bugs continuously, monitor how your customers react to new features in real time, and perform objective A/B testing to guide your product on solid facts rather than hunches or year-old marketing feedback. For instance, if you want to see whether a new feature can make your customers more productive, you can offer the corresponding release to a small subset of your customer base and examine changes to their work volume, response latency, click-through ratio, or any other objective measure that can help you decide.

The easy availability of cloud-based services and the associated network effects (the winner-takes-all phenomenon) result in a development infrastructure that's becoming increasingly homogeneous. Consequently, we developers can often use the same tools across diverse projects and employers. For instance, a budding programmer can first use GitHub as a student for a class assignment, then over the summer for contributing to an open source project, and later at work as a paid employee. This results in the transfer of knowledge and skills from one job to the next and a deeper talent pool of experienced developers for organizations. It also means that familiarity or expertise with a diverse set of services and technologies is no longer an optional skill but can be a hard job requirement.

DIOMIDIS SPINELLIS is a professor in the Department of Management Science and Technology at the Athens University of Economics and Business and the author of the recently published book *The Elements of Computing Style: 190+ Tips for Busy Knowledge Workers* (Leanpub/Lulu, 2014). Contact him at dds@aueb.gr.

See www.computer.org/software-multimedia for multimedia content related to this article.


IEEE Software
BRIDGING SOFTWARE COMMUNITIES THROUGH SOCIAL NETWORKS
IEEE Software
TWIN PEAKS OF REQUIREMENTS AND ARCHITECTURE
IEEE Software
SAFETY-CRITICAL SOFTWARE
IEEE computer society