

ON OUR
RADARIQT
IN-Q-TEL

LEVERAGING OPEN SOURCE SOFTWARE

By Troy M. Pearsall and Will Radosevich

At the time of In-Q-Tel's inception in 1999, the Internet bubble had reached epic proportions. There was a general acknowledgement that software product and platform innovations were no longer being directed at, or driven by, the government market.

This led to IQT's initial mandate to identify software innovations in specific areas such as natural language processing, geospatial analysis, and machine translation, among others, and adapt and deliver them for use by our government customers. The outcomes of this mandate are still used today on a daily basis by our Intelligence Community client agencies, as well as countless commercial entities and individuals. IQT achieved these results by engaging with the entrepreneurs that founded these companies, and the venture capitalists that fund them, to advance and commercialize their innovations. These efforts continue to fundamentally advance the Intelligence Community's software-enabled capabilities, while lowering its costs and extending its reach.

While IQT's mandate has expanded over the past ten years to also include physical and biological as well as security technologies, our focus on software and infrastructure endures. Given this continued focus,

it is necessary to understand and leverage the open source software movement presently underway.

The organizational construct of software development and production is changing. In the earliest days of computing, all software was custom built by the organization, often governments, which owned and operated the computer. As computers entered the business world, software was provided by the hardware vendor as part of an overall computing package designed for a specific purpose. Interestingly, source code was often provided in this model. With the rise of mini-computers and shared operating environments, software development and sales became the provenance of independent software vendors (ISVs) that often targeted a single platform or operating environment. The advent of the PC era spawned software companies that became household names: Microsoft (Word, Excel), Adobe (Flash, Reader), and, for those with slightly longer memories, WordPerfect and Lotus 1-2-3 (pre-IBM, pre-Notes). Each of these organizational constructs had one trait in common: They were (and in some cases, are) corporate entities that employed teams of software developers, directed their work, controlled access to the source code, and monetized the results often through software licenses. Software developed through this model of production is labeled "closed source."

In parallel with the evolution of the closed source software development model, a separate means of software production began to take shape. Tracing its roots back to Richard Stallman's GNU Project

and further brought into the mainstream by Eric Raymond's 1997 analysis of the Linux development methodology described in his essay, *The Cathedral and the Bazaar*, the open source software movement has established itself as an alternative to the closed source model of software development. A core tenet of the open source software movement is that recipients of software must be provided with source code under licensing terms that allows them the freedom to modify it for their own purposes. Over time, the open source software movement has also embraced the public development model, first popularized by Linux. That is, not only is source code provided to end users, but code development and delivery is accomplished through public forums. While not all open source software projects operate in this manner, many do.

So as a strategic investor, why is a shift in the means of software development and production of interest to IQT? In short, why do we care?

- The economics of software development are fundamentally changed. Cost is shifted from the corporate entity to the open software development community. The corporate entity becomes a caretaker of the community, building and fostering its growth. Not all companies will transition to this new role well, thus opportunities to disrupt incumbent providers will exist.
- Startup companies have a reach and potential impact that is disproportionate to their size. No longer are two persons in a garage a team of two. They, literally, could be managing a community of thousands of developers with contributors from around the globe. Open source and crowd sourcing methodologies are leveling the playing field in many industries.
- Open source development teams have created an infrastructure that supports global projects. Built up over the past decade, it is now possible to

coordinate self-organized teams worldwide with minimal or no direct expenses. This infrastructure is now being leveraged outside of the open source community to neutralize the advantages that normally accrue to large organizations.

- Every commercial and government entity will have to develop policies and procedures that govern the use of open source software within their organization. While we have been discussing open source software as if it were monolithic, the truth is that each license comes with its own set of rights, restrictions, and responsibilities. Effective and legal use of open source software will require that organizations understand the relevant licenses and the implication of choosing an open source solution.

The breadth of open source software projects has expanded considerably since Stallman founded the GNU Project. Early projects focused on development tools (gcc, Perl), operating systems (Linux, BSD), and web servers (Apache, nginx). Today there is an open source entrant in nearly every category of software, from office suites (OpenOffice, KOffice) to customer relationship management software (SugarCRM, vtiger CRM), statistical analysis packages (R, PSPP), and content management systems (Radiant, Drupal). As articles in this issue of the *IQT Quarterly* make clear, open source software has permeated every aspect of computing, powering the Internet's most highly trafficked websites, enabling the cloud, and changing the economics of software along the way.

One thing is certain: This shift in means of software development and production has caused a concomitant change in the entrepreneurial and investor communities. It will impact how, where, and when IQT can engage and foster the development of emerging capabilities delivered through software. This issue of the *IQT Quarterly* explores this evolving landscape, offers insight, and seeks to serve as a guide for further exploration. **Q**