IEEE Software
www.computer.org/software

Sprinting toward Open Source Development

Greg Goth

Vol. 24, No. 1
January/February 2007

This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

IEEE COMPUTER SOCIETY

© 2007 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

For more information, please see www.ieee.org/portal/pages/about/documentation/copyright/polilink.html.

Features Editor: Dale Strok ■ dstrok@computer.org

Sprinting toward Open Source Development

Greg Goth

During mid-2006, the open source project PyPy (http://codespeak.net/pypy/dist/pypy/doc/news.html) was garnering quite a bit of attention at developers' conferences. The project, which aims at developing a fast, flexible implementation of the Python language in Python, uses sprint-driven development. In sprints, a group of developers meet at one location and pair off for short periods (for example, two or three days) to develop part of a software project.

PyPy project manager Beatrice Düring says she was impressed by how many people wanted to talk to her about PyPy after her Agile 2006 and XP 2006 presentations. She says her husband jokingly told her that the attention was "because you're a woman!"

"I said, 'No, it's because it's a really sexy project!'" Düring says. "It's really interesting, because it has so many layers. It's interesting because of the global contextual view."

That contextual view might be of great interest to a wide range of developers and researchers. PyPy's hybrid development approach combines the distributed paradigm prevalent in free and open source software (F/OSS) development with sprints, which are usually associated with agile development and Extreme Programming. However, broader acceptance of such a hybrid approach might also mean that the community at large will need to reconsider siloed definitions of development methodologies.

"I know there's a formal definition of agile, but both agile and XP are terms that have become so morphed—because different people interpret them differently—that I hesitate to call what we do anything specific like that," says Allen Gunn. Gunn is the executive director of Aspiration Tech (www.aspirationtech.org), a San Francisco-based nonprofit agency promoting the use of open source software in the nonprofit community through developers' sprints and "convergences" of wider appeal. (Convergences are similar to conferences, but without panels or keynote addresses.) "But our events are definitely informed by the same set of values."

Agile values, nimble definitions?

Others have also noted the connection between agile and F/OSS methods. In the 2004 paper "When Is Free/Open Source Software Development Faster, Better, and Cheaper Than Software Engineering?" (www.ics.uci.edu/~wscacchi/Papers/New/Scacchi-BookChapter.pdf), Walt Scacchi succinctly explained the "sweet spot" in which practitioners might best examine the hybrid approach PyPy is taking, and from which other F/OSS projects might benefit:

> *F/OSSD projects are iteratively developed, incrementally released, reviewed and refined by software development peers in an ongoing agile manner. … These methods ensure adaptation to shifting user/developer requirements that are conveyed through informalisms. They also ensure acceptable levels of quality, coherence, and security of system-wide software via continuous distributed testing and profiling. … Agile software development practices are therefore closely aligned to F/OSSD practices, though it may be fairer to observe that agile software development*

0740-7459/07/$20.00 © 2007 IEEE

methods stand somewhere in the middle ground between SE and F/OSSD practices.

Düring says that, in the wake of the reaction to her conference presentations, perhaps the PyPy project's methodology itself shouldn't be the crux of the discussion. In fact, she says, even members of the PyPy team have disagreed on whether to call the project's methodology agile, which could alarm those with more structured definitions of the term. Also, the agile community might perceive the F/OSS community as claiming to be working agile while going against some of its core principles such as colocated development, Düring says. Rather, she says, the time might have come to widen developers' perceptions about methodological compatibilities and stumbling blocks as development grows more global and distributed:

"In some sense, by its nature, [sprint-driven development is] not very agile. When the agile community talks about a sprint they know exactly what they mean, so there could be a double territorial problem. Not only that we're using one of their established terminologies from one of their established processes [for instance, scrum projects hold a series of monthlong iterations called sprints], but also that more and more companies are looking at F/OSS and the way F/OSS is being developed. There are great learning opportunities, but also confusion."

Sprints nothing new in open source

The PyPy team's efforts to disseminate information about the project might help alleviate the confusion Düring mentions, in several ways. The documentation the team has assembled for its periodic EU milestone reports can offer concrete evidence of the efficacy of the sprint-driven approach within the F/OSS development model. In less quantifiable ways, higher visibility for the project might also engender broader discussion of how periodic onsite sprint sessions will evolve in a global economy ever more accepting of F/OSS products.

The sprint methodology in the F/OSS model isn't new with PyPy. It actually derives from a similar technique done by the Zope Corporation, which develops an open source content management system also based on Python. Zope chief technology officer Jim Fulton, writing in the February 2005 *ZopeMag* (www.zopemag.com/Issue010/Section_Articles/article_WhyZope3.html), says that the initial Zope XP sessions took place late in 2001 as the company began work on Zope 3. Zope staff member Tres Seaver had been advocating using XP methods at Zope, and Fulton named the first short development session a "sprint."

"In December 2001, we announced the Zope 3 project to the Zope Community as we were in the process of opening the Zope repository to contributors outside of Zope Corporation," Fulton writes. "Zope 3 provided a major opportunity for people to get involved in the Zope development process. We promoted the idea of sprints and held 3 Python sprints in Fredericksburg (Virginia), in January and February of 2002. To date, there have been 28 community Zope 3 sprints. Sprinting has turned out to be great for

- involving people in the Zope 3 development process,
- building the community, and
- completing important work.

The most important benefit of sprints is allowing people to meet, get to know each other, and collaborate in person. This makes it much easier subsequently to work remotely."

Consultant James Shore is considered a leading practitioner of agile methodology. In fact, he won the Agile Alliance's 2005 Gordon Pask Award for his contributions to the agile community. Shore, who has also led two open source projects, concurs with Fulton's sentiments that, rather than being mutually exclusive, face-to-face development gatherings within open source projects are incredibly beneficial.

"There is just something that happens when people get together in a room that doesn't happen when they work long distance," Shore says. "One of the big challenges with F/OSS proj-

IEEE Software

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
Access www.computer.org/software for information about *IEEE Software*.

Subscribe
Visit www.computer.org/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, send email to 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.

ects is not the fact they're F/OSS—what is relevant is most F/OSS projects have people working at long distance and not necessarily able to give it their full attention. That's a big challenge for keeping everybody on the same page, and I think sprints are a fantastic innovation for making that happen."

However, Shore contends that the PyPy project veers far enough away from the four-point agile manifesto (http://agilemanifesto.org) that it shouldn't be called agile per se.

"I think it would be a mistake to say this is a combination of agile and open source development," he says. "I think it has applied one of the ideas of agile development to the very difficult problem of distributed software development."

However, trying to divine what is or isn't agile development for an F/OSS sprint-driven project might be a matter of several variables (such as frequency of sprints), and applying formulaic productivity measurement might be difficult to impossible. Part of the PyPy team's philosophy holds that the method they have chosen works for them; they also strongly caution that other project teams should carefully consider their technological and procedural goals before beginning a similar approach.

PyPy success factors

The PyPy team's most recent milestone report for the EU (http://codespeak.net/pypy/extradoc/eu-report/D14.3_Report_about_Milestone_Phase_2-final-2006-08-03.pdf) mentions several success factors that could serve as guidelines for similar projects' success. Some of these factors are technical and some methodological or procedural.

Technologically, for example, the number of lines of code in PyPy has grown consistently from about 40,000 when the project began monitoring in 2004, concurrent with EU funding, to 250,000 in July 2006. Test code has increased at a corresponding rate, now totaling about 60,000 lines of code. Also, the 0.9 PyPy release in June 2006 achieved a 20-fold performance improvement over the preceding release through a series of static and core translation optimizations.

Procedurally, the PyPy group's dissemination strategy has melded its steady technical progress with sprints held in diverse areas of the world. According to the milestone report, both commercial and academic groups either participated in PyPy sprints or hosted them. Among the pioneering collaborations was a sprint hosted by the Advanced Institute of Science and Technology in Japan, which was "piloting their interest in hosting F/OSS events," according to the milestone report. The PyPy group was also hosted by the computer science department and local F/OSS group at Palma University in Mallorca, Spain. Commercially, companies including IBM, Hewlett-Packard, Iona Technologies, and Canonical have expressed interest in PyPy. Some commercial entities have participated in sprints, while others have expressed interest in holding more general PyPy workshops. The number of developers actively following PyPy changes has also steadily increased since the EU funding began. As of June 2006, 50 developers were actively following changes to the code base, while the developer mailing list has doubled, from 150 to 300.

Düring says that the project leaders plan to simplify access to PyPy's core data for those interested in researching its methodology. Such detailed empirical data might indeed be useful for public-sector funding organizations or commercial entities laden with due-diligence mandates from shareholders or regulators. However, sprints are being increasingly used in far less quantifiable contexts, and their organizers show no sign of slowing their sprint rollouts.

Developers and project managers interested in sprints' suitability need to look at many projects using them, to develop a sound development scaffold.

Other sprints look for the x factor

"At our sprints, we think much more about how to build team collaboration instead of how to build a specific feature," Aspiration Tech's Gunn says. "To the extent the Internet has isolated us by enabling us to collaborate via email or cell phone, the role of the event is in some sense becoming more central. There is less face-to-face meeting in an email-driven world. Conferences are becoming more ritualistic. It used to be you went to a trade show to see what everybody was doing. With the Internet, you already know what everybody is doing, and now the emphasis has shifted to the importance of conversations in the hallway, talking about families and grabbing a beer. That human element is a critical x factor in building successful collaborations."

Gunn says that traditional productivity metrics have trouble capturing that x factor. Aspiration's mission isn't to demonstrate to a corporation or consortium that a given project is worth funding. Instead, its mission is to show nonprofit agencies how OSS can help them advance their cause and to help the projects' developers hone their technology for a niche market that fits with the F/OSS ethos.

"We don't apply concrete metrics very often to our events, because for what we do, it would be kind of arbitrary," he says. "Having been a software engineer as long as I have, I am not at all enamored of things like lines of code or module counts. … We measure things in terms of testimonials, things like developers talking about the impact our events have had on their project."

Gunn says Aspiration's sprints have included hybrid sprints such as usability sprints that bring F/OSS developers and human-computer interface experts together to work on iterative design improvements.

"At the usability sprints, the emphasis is on teaching developers processes to enhance their projects' usability, and the coding piece is almost like the punchline," Gunn says. "They focus on the process of identifying a usability goal, defining a test process, applying analysis to their tool, and then get to coding."

The usability sprints' goal, he says, is to spend the two days addressing one usability issue per tool, "driving a process to get resolution on that one issue. I would say a third to a half of those relationships [between OSS developers and the nonprofit customers they meet at the sprints] prosper. The other half, I think, [fail owing to] a combination of things, the main mitigating factor being everybody goes home to a very busy world."

Aspiration also holds strictly development-focused sprints such as a voice-over-IP application sprint. That, Gunn says, was "three days of nonstop 24/7 coding, and that was much more in the agile model where we had target users telling developers what they needed, we had coders coding, and people testing more or less in real time."

Gunn's assertion that developers can spread the word of sprints' success via testimonials is borne out by Mimi Yin, an Open Source Applications Foundation designer, in her blog postings (http://wp.osafoundation.org/author/mimi). After the August 2005 usability sprint, Yin wrote,

> *I felt like we accomplished an amazing amount of work in a very short period of time and gained a lot of visibility into a workflow that was foggy at best at the beginning of the three days. … The FLOSS usability sprint opened my eyes very wide as to quite how extremely rapidly you can iterate on design.*

Last lessons from PyPy

Over the project's life, Michael Hudson, one of the PyPy team's core developers, has gained perspective regarding the benefits and challenges of sprint-driven development within PyPy.

"Without that very first sprint, it's quite possible [the PyPy project] could have been just one of those ideas that emerged for a week on a mailing list and then disappeared without achieving anything," Hudson, a scientific collaborator at Heinrich Heine Universität, says. "It's a big, complex project, and without sitting down in the same room, we might never have been able to communicate what we needed to."

While Hudson asserts that sprint-driven development such as PyPy's is a growing trend among F/OSS projects, he adds that external economic factors can play a huge role in determining whether sprints will work. For example, he says that PyPy probably would have withered away without factors such as cheap airfare and affordable laptop computers in the project's early days, when it was entirely self-funded.

At the same time, given the size, complexity, and financial underpinnings of PyPy's development, Hudson mentions project aspects that present real challenges to maintaining both the agile paradigm and the coherence of vision surrounding a successful F/OSS project.

"It's just hard work," he says. "If you're going to sprints every six weeks, it does take time and attention from other aspects of life. From time to time, there are also differing goals among developers from different companies, and periodic changes in priorities. There is an EU process that allows for changing the development plan, but it's pretty heavy handed."

Those issues aside, Hudson says he's quite happy with the PyPy experience and would try a similar approach if he got the chance to work on another such project.

"The amount of technical progress in PyPy has been quite amazing," he says. "I love to be paid to work full time on an open source project."

Düring says developers and project managers interested in sprints' suitability need to look at many projects using them, to develop a sound development scaffold.

"The uniqueness of PyPy is, we're budgeted for 14 sprints every two years," she says. "Everything is designed to include the process around sprints. But don't forget to look at existing examples of sprint-driven development such as Plone [http://plone.org] and Canonical [www.canonical.com], which typically hold sprints two to three times a year. There are different contexts organization-wise, geographically, etc.

"When you look at the agile community, one big subject of papers is how to cram agile practices into global-development logistics, and a big question is, can you even call sprints agile, because [they go] against some of the core principles.

"But distributed development is a fact, and it's shaping the world of software development in terms of tools and how we handle version control, and it's growing. Look at some of the more interesting techniques out there. Look at contexts."