Open Source Software in an Agile World

Steven Fraser[1], Pär J. Ågerfalk [2], Jutta Eckstein[3],
Tim Korson[4], and J.B. Rainsberger[5]

[1] Senior Staff, QUALCOMM, San Diego, USA
sdfraser@acm.org
[2] Research Fellow, University of Limerick, Ireland
par.agerfalk@ul.ie
[3] Consultant, Objects in Action, Braunschweig, Germany
jutta@jeckstein.com
[4] Consultant, Korson Consulting, TN, USA
tim@korson-consulting.com
[5] Founder, Diaspar Software Services, Toronto, Canada
me@jbrains.info

Abstract. Open Source Software (contrasted with proprietary or "closed" software) has become a more widely accepted enterprise solution not withstanding some issues related to intellectual property rights and issues of liability and indemnification. Open Source Software (OSS) takes collaborative software development to a global extreme – OSS also provides a mechanism for decreasing time-to-market, improved quality, and reduced development costs. This panel will serve as a catalyst to discuss strategies, tools, and communities focused on the development and application of open source software.

1 Steven Fraser (Panel Moderator)

Steven Fraser recently (January 2005) joined QUALCOMM's Learning Centre as a member of senior staff in San Diego, California – with responsibilities for tech transfer and technical learning. From 2002 to 2004 Steven was an independent software consultant on tech transfer and disruptive technologies. Previous to 2002 Steven held a variety of software technology program management roles at Nortel and BNR (Bell-Northern Research) - including: Process Architect, Senior Manager (Disruptive Technology and Global External Research), Advisor (Design Process Engineering), General Chair (Nortel Design Forum), and Software Reuse Program Prime. In 1994 he spent a year as a Visiting Scientist at the Software Engineering Institute (SEI) collaborating with the "Application of Software Models Project" on the development of team-based domain analysis techniques. Since 1994, Steven has regularly moderated panels at ACM's OOPSLA and other software conferences – serving as OOPSLA panels chair in 2003 and as XP2006's General Chair. Steven holds a Doctorate in Electrical Engineering (software graphics standards validation) from McGill University in Montreal, Canada, an MS in Physics (Queen's University at Kingston), and a BS in Physics and Computer Science (McGill University). Steven is a member of the ACM and IEEE.

P. Abrahamsson, M. Marchesi, and G. Succi (Eds.): XP 2006, LNCS 4044, pp. 217 – 220, 2006.

2 Pär J Ågerfalk

The open source software (OSS) is a global phenomenon with developers spread across the world. At the same time, the OSS model is an agile approach that manages to adapt fluently to changing situations and which is known for producing high-quality code with swift handling of the few bugs that remain in released software. In the proprietary world, the understanding of agile approaches in global software development is still quite limited. Hence, understanding better the interplay between agile methods, OSS and global software development is an important topic that should benefit all three 'communities' (OSS, agile and commercial/proprietary). In this panel I will draw on recent research in this area and present a number of challenges that should be part of a research agenda for the intersection of OSS and agile methods.

Pär J Ågerfalk is a research fellow at the University of Limerick and an assistant professor (universitetslektor) in informatics at Örebro University, where he heads the Methodology Exploration Lab. He received his PhD in information systems development from Linköping University in 2003. His research on systems development method flexibility, language/action based information systems theory and open source software development has resulted in more than 50 publications in a variety of journals, books, and international conferences and workshops. He has served on the committees of numerous conferences and is an associate editor of European Journal of Information systems as well as of the electronic journal Systems, Signs and Actions (www.sysiac.org). Ågerfalk is scientific manager and deputy coordinator of the EU FP6 Co-ordination Action project CALIBRE (www.calibre.ie), co-leading the distributed development work package and coordinating the scientific side of the CALIBRATION open source industry research forum. He was the lead author of the paper 'Assessing the Role of Open Source Software in the European Secondary Software Sector: A Voice from Industry', which won a best paper award at the 1st International Conference on Open Source Systems in Genoa 2005.

3 Jutta Eckstein

Open Source provides a great leverage for implementing the fist value: Individuals and interactions over processes and 'tools'. I regard this as the major reason why I have never seen an agile project without using any kind of Open Source software. Using the official purchasing department in order to acquire a new tool takes typically too long to provide the necessary quick feedback an agile team needs. On the other hand Open Source software development, although being distributed, implements a lot of agile techniques, sometimes even the agile value system. This provides for commercial agile teams great learning opportunities. Thus agile software development and Open Source form a give-and-take relationship.

Jutta Eckstein is an independent consultant and trainer for over ten years. She has a unique experience in applying agile processes within medium-sized to large mission-critical projects. This is also the topic of her book Agile Software Development in the Large. Besides engineering software she has been designing and teaching OT courses in industry. Having completed a course of teacher training and led many 'train the

trainer' programs in industry, she focuses also on techniques which help teach OT and is a main lead in the pedagogical patterns project. Jutta has presented work in her main areas at ACCU (UK), OOPSLA (USA), OT (UK), XP (Italy and Germany) and Agile (USA). Jutta is a member of the board of the Agile Alliance and a member of the program committee of many different European and American conferences in the area of agile development, object-orientation and patterns.

4 Timothy Korson

I am not an open source zealot, not am I a strong proponent of any particular commercial software environment, but I do care passionately about the process of building software better, faster, and cheaper. From this perspective I believe that both the Agile community and the Open Source community have given us valuable insights about how to develop software. And these are not just theoretical insights. Both communities have demonstrated to us practical techniques that work. For example the XP concepts of pair programming and shared ownership are really taken to the extreme in the open source community. These and many other lessons are there for all of us to learn and apply in our own companies if we but have the courage to try.

Timothy Korson has had over two decades of substantial experience working on a large variety of systems developed using modern software engineering techniques. This experience includes distributed, real time, embedded systems as well as business information systems in an n-tier, client-server environment. Korson's typical involvement on a project is as a senior management consultant with additional technical responsibilities to ensure high quality, robust test and quality assurance processes and practices. Korson has authored numerous articles, and co-authored a book on Object Technology Centers. He has given frequent invited lectures at major international conferences and has contributed to the discipline through original research. The lectures and training classes he presents are highly rated by the attendees.

5 J.B. Rainsberger

I have primarily been a consumer, rather than a producer, of Open Source tools, libraries and software. As a general computer user, I rely on Open Source tools for most of my basic computing needs: e-mail, browsing, personal organization, word processing, spreadsheets. The abundance of free general-purpose tools makes it easy for me to pay for more specialized software, so I can support fellow software professionals better. In my role as a software developer, I rely on Open Source tools for my development platforms. Having used many expensive development environments, it is obvious to me that the Open Source community produces superior work overall, since in many cases, I find high defect rates much more costly than smaller feature sets, and commercial environments tend to deliver more defects in hopes of delivering more features than Open Source projects. Also, while companies believe themselves to be under continuous time pressure to deliver, the best Open Source projects tend to release more frequently, with rich feature sets and low defect rates.

I suppose it's true what “Peopleware" by DeMarco and Lister says: “teams allowed to set their own deadlines often finish sooner”.

In my work as a consultant and programmer, I emphasize reusing existing libraries as a way to counter the Not invented here attitude that afflicts many software teams. I have already seen considerable improvement, as teams that build on existing work tend to learn more about what's possible for their project than those who build more for themselves. Programmers spend very little time reading code, so reusing Open Source libraries gives them an excellent opportunity to do just that. In so doing, they learn much more about their platform, about what makes good and bad design, and about what features are possible. It is a simple way to expand the team in some sense to include considerable outside expertise. While occasionally we run into libraries we wish we'd never found, I never consider that time wasted, as it sharpens each person's understanding of what makes a good or bad product. The Open Source community provides an invaluable service to those who deliver software for a living, and even to those who simply use computers on a regular basis. We owe them much for their efforts.

J. B. Rainsberger is the Founder of Diaspar Software Services, where he coaches both individual programmers and entire teams in value-driven software development practices. His book, JUnit Recipes is the top-selling book for Java programmers about JUnit, testing and test-driven development. Joe has been an XP practitioner, researcher, presenter, and author since 2000 – and in 2005 received one of the first Gordon Pask awards for contribution to Agile practice.