Mind Soc (2013) 12:113–123
DOI 10.1007/s11299-013-0125-5

Free and open source software (FOSS) as a model domain for answering big questions about creativity

Scott Dexter · Aaron Kozbelt

Received: 27 June 2012 / Accepted: 22 February 2013 / Published online: 8 March 2013

Abstract In free and open source software (FOSS), computer code is made freely accessible and can be modified by anyone. It is a creative domain with many unique features; the FOSS mode of creativity has also influenced many aspects of contemporary cultural production. In this article we identify a number of fundamental but unresolved general issues in the study of creativity, then examine the potential for the study of FOSS to inform these topics. Archival studies of the genesis of FOSS projects, coupled with laboratory studies detailing the psychological processes involved in software creation, can provide converging evidence on the nature of creativity in software design. Such a research program has broad implications both for theories of creativity and for real-world innovation in software and other forms of digital cultural production.

Keywords Creativity · Software · Free software · Open source software · Collaboration

1 Introduction

The increasing prevalence of free and open source software has been a major development in software engineering over the last 15 years (DiBona et al. 1999, 2005; Weber 2004; Feller et al. 2005; Chopra and Dexter 2007). Users worldwide have eagerly adopted software developed using FOSS, such as the GNU/Linux

S. Dexter (✉)
Department of Computer and Information Science, Brooklyn College,
2900 Bedford Avenue, Brooklyn, NY 11210, USA
e-mail: sdexter@brooklyn.cuny.edu

A. Kozbelt
Department of Psychology, Brooklyn College, 2900 Bedford Avenue, Brooklyn, NY 11210, USA
e-mail: aaronk@brooklyn.cuny.edu

operating system, which undergirds many sensitive computing-intensive corporate and scientific computing installations; the Apache server, one of the most widely deployed Web servers; and Mozilla Firefox, a popular web browser. Moreover, FOSS is an increasingly common component of commercial software development (Capra et al. 2011), a development that impacts the cultures of both corporate participants and the FOSS community (Lin 2006).

In this paper, we argue that FOSS is a novel and useful domain for the empirical study of creativity. Not only has FOSS been closely aligned with recent changes in cultural and technical production (Benkler 2006; von Hippel 2006), but its inherent openness provides a remarkable trove of data which may yield new evidence bearing on the nature of creativity. Below, we briefly summarize the important characteristics of the free and open source software domain; in subsequent sections we point to some gaps in the current state of our understanding of creativity, then discuss the potential of the study of FOSS to close these gaps.

Unlike proprietary software, which tends to be generated in corporate environments and whose code is a closely guarded trade secret, FOSS projects are designed to invite voluntary contributions: programmers can modify an existing public codebase and share the modified code with others, without restriction (Free Software Foundation 2012). However, FOSS is more than a unique software development ethos: its characteristics and mores continue to shape many broader aspects of society and culture. Scientific practice is increasingly dependent on software, and the transparency of this software is understood more and more as a crucial element of modern scientific peer review (Barnes 2012). FOSS is increasingly prominent in discussions about openness and transparency in e-government, as in the recently adopted European Interoperability Framework (European Commission 2010) for European public services. And of course FOSS is closely tied to various aspects of the "free culture" movement, including Creative Commons licensing (Lessig 2004; Doctorow 2008).

FOSS further possesses a number of characteristics that make it uniquely suitable as a domain for the study of creativity, as we shall see below. Because of its self-organizing dynamic, FOSS permits and demonstrates a variety of modes of collaboration in the service of collectively developing the code. Perhaps more importantly, because software development (and FOSS in particular) relies heavily on the existence of a full archive of both the code and developers' discussions of it, it is possible to characterize, very precisely, the state of the code at any point in its history. That is, FOSS offers a very rich 'paper trail' of the evolution of the code itself, including 'meta' discussion about changes on mailing lists, on code submissions, and/or in comments embedded in the code. Thus, FOSS projects, individually and collectively, present a rich source of longitudinal data against which hypotheses about the collaborative and creative processes of software development may be tested.

2 Several open questions in the study of creativity and innovation

Creativity has been the subject of serious scientific investigation only for about the last 60 years (see Guilford 1950), with a notable acceleration in the last two

decades. Scholarship in this area has yielded some useful initial results and a general consensus on key constructs and methods (Kaufman 2009). Given the complexity of the phenomenon of creativity, many fundamental issues remain unresolved. How is creativity related to intelligence (Kim et al. 2011)? Mental illness (Silvia and Kaufman 2011)? Personality (Eysenck 1993)? What are the patterns of neurological activity during creative or insightful thinking (Kounios and Beeman 2009)? What is the nature of social judgments of creative products (Amabile 1983; Sawyer 2006)? Beyond these basic psychological and measurement questions, here we focus on issues related to the structure and nature of the creative process—issues which the study of FOSS seems most likely to inform.

Many current models suggest that creative cognition is distributed across different regimes, or stages, of thought (e.g., Getzels and Csikszentmihalyi 1976; Martindale 1990; Simonton 1984; Ward et al. 1999). Typically, one of these stages involves generating ideas; the other, elaborating them into finished products. However, no extant models provide much in the way of a detailed characterization of the structure or dynamics of either aspect of the creative process. Compounding this impasse is the generic, domain-general quality of such "stage" models (Baer 2011), which fail to incorporate any specific content of the domain in which a creator works—such as oil painting, subatomic physics, or software development using a particular programming language.

These limitations aside, theories of stage models have yet to determine the relative importance of ideation versus elaboration. Most theories show a strong bias toward free-associative ideation as *the* engine of creativity, de-emphasizing the value of more conscious elaborative processes (Kozbelt 2009). Such a view not only distorts the phenomenon of creativity but also implies that at root the creative process is not amenable to any kind of deliberate control (see Simonton 1999). However, mounting evidence (e.g., Kozbelt 2006, 2008; Silvia 2008) suggests that creators *are* able to exercise considerable control over aspects of the creative process. They show reliable discernment of ideas and evaluation of in-progress and finished works, and they produce longitudinal improvement in the quality of their creative productions. Precisely how this happens is poorly understood, but the possibility of meaningful creative control raises significant challenges to many current conceptualizations of creativity. Rising to such challenges will likely require far more—and more specific—data than have hitherto informed most studies of creativity.

Fortunately, an abundance of domain-specific data spanning the entire sweep of the creative process, which potentially informs these unresolved issues about creativity, is a natural consequence of FOSS practice. Understanding why this is the case requires an understanding of some particulars of FOSS, to which we now turn.

3 Empirical studies of FOSS

The unique character of FOSS relies on two technical features: legal licenses and software development tools. FOSS licenses—which specify the kinds of actions that programmers and users are permitted, mandated, or forbidden to take—have a

variety of forms (Lerner and Tirole 2005; Rosen 2004), but all generally agree that all users must have access to the source code and must be permitted to modify and/or redistribute this source code. With these properties, FOSS becomes the kind of software which may be studied, modified, and improved by an unconstrained community of interested persons: the "bazaar" of the seminal 2001 *The Cathedral and the Bazaar* (Raymond 2001). This community is supported and enhanced by a suite of tools for distributed communication, including mailing lists, online databases for tracking problems with the code, and, most importantly, source code management systems which substantially automate the collection, integration, and redistribution of changes to the code.

The FOSS model, then, treats software as a work in progress: every proposed modification to the code will be accepted or rejected—and always recorded—by the community. A project's developmental trajectory, propelled by the complex and diverse decision-making processes of the developers, is similarly committed to a publicly-visible archive. Such public records collectively represent an enormous trove for empirical study from a variety of perspectives.

In several important respects, then, FOSS differs from domains that have been prototypical for the study of creativity, such as visual art (e.g., Getzels and Csikszentmihalyi 1976) or scientific discovery (e.g., Langley et al. 1987). First, in FOSS, the creative process invariably involves *socially distributed cognition* (Hutchins 1995). Some recent perspectives on creativity (e.g., Sawyer 2006) have emphasized distributed cognition in contexts like jazz and improv comedy, but FOSS differs even from these collaborative domains in that participants need not be in close proximity or even direct communication with each other. Second, the public and heavily-instrumented nature of FOSS development affords intensive longitudinal analysis, ranging from the reconstruction of the state of the code at any given point, to discourse analysis of developers' dialog, to data-driven empirical analysis of many factors pertaining to the code's development. Finally, while code is certainly assessed on subjective aesthetic grounds, as the essays in Oram and Wilson (2007) show, code quality can also be assessed on any number of empirically derivable metrics (Antoniades 2007).

Such code quality metrics tell us much about what software developers take "good" software to be. Because software objects are generally useful objects with long useful lives, they are far from static. Developers' creativity, then, must be aligned with the need for future modifications and corrections to the software. In essence, they must strive to manage software's complexity, though such complexity manifests in a variety of ways. For example, a foundational work (Chidamber and Kemerer 1994) on software metrics for "object-oriented" software, the dominant industrial paradigm, introduces metrics such as Weighted Methods per Class, which is "a predictor of how much time and effort is required to develop and maintain the object" (482), and Lack of Cohesion in Methods, which "increases complexity, thereby increasing the likelihood of errors during the development process" (489). Metrics such as these are both objective and multi-dimensional in their assessment of code quality, and also can be applied to in-progress states of the code. The potential application of objective metrics to a rich repository of evolving creative artifacts means that limitations of reliability and validity, which are traditionally

associated with subjective ratings in the creativity research literature (Amabile 1983), can be largely circumvented.

FOSS project archives have been the subject of several substantial empirical investigations seeking to understand something of FOSS social dynamics. Perhaps the first such study, the Orbiten Free Software Survey (Ghosh and Prakash 2000) focused on extracting some basic data about authorship and size from a relatively small set of FOSS projects. Much investigation has been focused, from an economic perspective, on the motivations of free software developers, who appear to work without the usual modes of incentive and compensation (Lakhani and Wolf 2005; von Krogh et al. 2012). Other studies have focused on questions of software engineering. For example, Mockus et al. (2005) conducted detailed case studies of two major FOSS projects, Apache and Mozilla, using specially-developed automated tools to extract relevant data from every electronic archive available. This process yielded valuable data about the development process, such as the number of individuals actively involved in various roles in the software's creation, the scope of each individual's work, the extent to which files were modified by several developers simultaneously, and the time required to solve various problems.

More relevant to our interests in the study of creativity, the European Commission recently allocated substantial funding to an array of related projects related to the "[i]nvestigation into the use of open source models for improving software engineering…. based on agreed indicators of productivity and quality" (European Commission 2006). For example, the FLOSSMetrics project, funded under this mandate, focuses on an improved understanding of "[the FOSS] development process, its productivity and the quality of its results," (FLOSSMetrics Consortium 2010). Most FLOSSMetrics studies had to do with software engineering concerns, such as maintainability, complexity, efficiency, and productivity. But some of these studies focused on process-related inquiries, such as the characterization of the evolution and dynamics of software and flaws in the software (Koch et al. 2009). In the next section, we argue that such process-related measures of FOSS projects can substantially inform psychological aspects of creativity more generally.

4 How FOSS can inform the nature of creativity

Our broad argument is that FOSS represents an ideal test bed for developing and testing theories that can fill in important lacunae in research on creativity identified above. FOSS projects have inherently high ecological validity, of course; more significantly, the thorough documentation of the evolution of such projects allows very fine-grained dynamic analyses of code quality. The nature of the fluctuation in the quality of an emerging creative product over time bears strongly on the question of whether creativity can be meaningfully controlled, with strong implications for characterizing the underlying psychological mechanisms (e.g., the extent to which the mechanisms are chance-driven vs. expert knowledge-driven: see Kozbelt 2008). Moreover, in the case of FOSS, the psychological richness of the documentation is

further enhanced at the 'meta' level by the comments and discussions of the contributions.

In this section, we sketch some details of how such a FOSS-centered research program might work, focusing on: (1) disentangling ideation- versus elaboration-intensive accounts of the creative process; (2) using quality trajectories of evolving code as an analogue to psychological studies of 'feeling of warmth,' which can inform the extent to which the creative process is subject to meaningful control; and (3) complementing an archival approach to FOSS studies by more controlled laboratory studies of programmers at work.

Previous research has already informed some of these themes in rudimentary ways. For instance, Mockus et al. (2005) examined the quality of code in the Apache and Mozilla projects by counting post-release defects per thousand lines of new code as well as defects per thousand changes to the code. They found that the defect density of the code prior to release is much lower for FOSS products compared to proprietary products. Possible causes, they suggested, are: (1) fewer defects are injected into FOSS; (2) that inspections are conducted more frequently or efficiently; or (3) FOSS developers are also users of the software and have considerable expertise in the domain. Specifically, FOSS contributors may be particularly astute at evaluating the quality of their own and others' code modifications, perhaps due to their high levels of motivation and considerable domain expertise (Sonnentag et al. 2006), or because experience in the FOSS domain encourages development of these skills.

Such findings, with their provisional explanations, have strong psychological overtones. Not only do they imply that FOSS programmers have high motivation and substantial expert knowledge; they also suggest that FOSS participants are collectively adept at evaluating their emerging codebase and guiding the process of creation to a good end. However, to date, archival studies that have examined FOSS projects have rarely focused on understanding psychological processes or the fundamental nature of the creative process. We now describe three ways in which FOSS can inform basic issues in the study of creativity.

4.1 Ideation versus elaboration

Because FOSS projects are continually propelled by input from various contributors, they provide a means of assessing the extent to which basic ideational versus elaborative mechanisms drive the process of creation. Put another way, do the major features of a creative outcome largely fall into place early on, with only fairly minor tweaking later, or is there a more gradual course of development? Creativity researchers sometimes speak of 'delay of closure' as a key aspect of the creative process; for instance, Getzels and Csikszentmihalyi (1976) found that in an open-ended drawing task, more creative art students took more time for the gestalt of the final composition to emerge than did less creative art students. Alternatively, more ideation-intensive accounts of creativity (e.g., Simonton 1999) tend to emphasize initial moments of insight as largely determining the course of the creative process.

These accounts have significant practical implications in the context of FOSS, as well as more generally. Pragmatically, this issue revolves around how resources

should be allocated throughout the duration of a creative project. An ideation-intensive view suggests one should devote the most resources at the start of the creative process, casting about widely for an intrinsically powerful idea (after which its development may be handled perfunctorily). An elaboration-intensive view would suggest that such a dynamic is counterproductive; instead, the development of a creative product should involve continual investment of resources as the product develops. Characterizing the extent to which ideas are introduced and modified, and the level of activity of revising code throughout the life of a FOSS project, would begin to inform this issue in a data-driven way. Naturally, there may be room for both perspectives across different FOSS projects or as a function of individual differences (see Kozbelt 2008). But regardless of the conclusion vis-à-vis the ideation-elaboration debate, a rich descriptive account of the relative roles of ideation and elaboration in the FOSS creative process would be useful in its own right, particularly given the dearth of such models in any domain, as noted earlier.

4.2 Quality trajectories, "feeling of warmth," and creative control

Data from FOSS projects bear on another central issue in the study of creativity: the extent to which the creative process can be controlled. FOSS provides a data-rich means of addressing this question by dynamically tracking fluctuations in quality as the code evolves. When objective software metrics are applied to a series of in-progress states of source code for a particular project, the shape of the quality trajectory for that code can be determined. Such trajectories are related to cognitive psychological studies of 'feeling of warmth,' which have been used as a way of differentiating insight versus incremental problem solving. In many such studies, problem solvers subjectively rate their perceived closeness to a solution (Metcalfe and Wiebe 1987; Jaarsveld and van Leeuwen 2005); in other studies of creative problem solving in art (Kozbelt 2006; Kozbelt and Serafin 2009; Serafin et al. 2011) other raters provide assessments of in-progress states of the artworks. The latter set of studies has yielded some informative findings. For instance, quality trajectories tend to show improvement over time and end at a quality level at least as high as any earlier state. Moreover, trajectories are frequently quite jagged, especially when leading to outcomes that are ultimately judged as higher in quality. These results imply that even in a complex creative dynamic, creators have fair perspicacity in knowing when to stop working—and thus can exercise some meaningful control over the process of creation.

While these findings shed some important initial light on these topics, arguably FOSS has a far greater potential to do so, given the availability of numerous objective software metrics, which circumvent the lower reliability and questionable validity associated with subjective ratings (see Amabile 1983). Moreover, in FOSS environments, such instances of significant discontinuities (positive or negative) in objective code quality can be related to the content of programmers' commentary and discussion. Thus, the richness of documentation inherent to FOSS practice can also uniquely inform the nature of the cognitive, metacognitive, and evaluative processes undergirding creative activity.

4.3 Laboratory studies

Building on this last point, additional insight on the mental processes that help propel FOSS projects forward can be garnered from laboratory studies of programmers. One powerful method is that of concurrent verbal protocol analysis, in which participants engage in a problem solving or creative task and verbalize their conscious thoughts while working (Ericsson and Simon 1984). Such studies allow detailing of the nature, frequency, and context of various categories of cognitive processes such as goals, plans, and evaluative statements (see Fayena-Tawil et al. 2011) at different points in the creative process. Such analysis may provide converging results with archival studies of FOSS code repositories. Beyond verbal protocol studies, survey studies can also target particular issues, such as the degree of software developers' reliance on aesthetic judgment during software creation. Indeed, a recent survey investigation (Kozbelt et al. 2012) suggests that this is a promising direction of study. Specifically, both expert and novice programmers reported having aesthetic experiences with code, though somewhat less frequently and intensely than with other creative artifacts. Participants also reported that judgments of "ugly" code were reported to be faster than those of "beautiful" code, which in turn were faster than those of "correct" code. Such results suggest that aesthetic intuitions may be a valuable aspect of problem-solving as programmers write and revise code.

5 Toward a richer understanding of creativity and innovation

The aesthetic and creative aspects of software development are not always easy to grasp without training in software engineering, but programmers themselves seem to find creativity- and aesthetics-laden language a natural way to describe their work (see accounts in Oram and Wilson 2007 and Lammers 1986/2006). The differences between creativity in the fine arts and in a domain such as software development are far from fully understood, but there are few domains other than FOSS which can offer large amounts of high-quality, ecologically valid, quantitative and qualitative data about the creative process.

FOSS-style creativity is, as well, increasingly at the center of mainstream and scholarly attention. Lawrence Lessig's *Free Culture* (2004), Erik von Hippel's *Democratizing Innovation* (2005), and Clay Shirky's *Cognitive Surplus* (2010) each express hope and concern for the growing possibility of valuable creative work being carried out by distributed groups. Simultaneously, a large research literature on group creativity continues to expand, emphasizing group dynamics and practical issues surrounding the effective implementation of novel ideas (e.g., Govindarajan and Trimble 2010). Notably, some of the best-supported contemporary research on innovation speaks quite directly to FOSS practice. For instance, the Motivated Information Processing in Groups model (De Dreu et al. 2008) posits that higher levels of creativity and innovation arise in groups whose members have both epistemic motivation (they systematically process and disseminate information) and prosocial motivation (they seek collective rather than personal gain). Both features

are characteristic of FOSS practitioners, who engage in extensive documentation and discussion of the code (thus showing high epistemic motivation), and who demonstrate a wide range of motivations (von Krogh et al. 2012), many of which are highly prosocial.

While we are optimistic about the prospects for learning much about the nature of creativity through the study of FOSS, the approach outlined here is not without its limitations. Chief among them is that the close study of FOSS creativity may simply not inform creativity in other domains, or creativity in more general, abstract terms. This trade-off between incorporating details of the content and processes of a domain versus lack of potential generalizability is an inherent tension in domain-specific models of creativity (Baer 2011). Another possible limitation concerns the potential variability among different FOSS projects: results from one project may suggest one conclusion, while results from another project may suggest a different conclusion. However, this would not necessarily invalidate the enterprise of trying to use FOSS to inform creativity, since in other domains, seemingly contradictory conclusions from different sub-samples can often be folded into theoretically unified accounts (Kozbelt 2008). Indeed, the great variability among FOSS projects may ultimately allow researchers to distinguish differences in the characteristic processes and dynamic quality trajectories of projects leading to, say, high- versus low-impact outcomes. Such a research strategy would emphasize more value-laden criteria that relate strongly to real-world concerns about creativity.

In conclusion, then, FOSS is (1) an aesthetic intervention into the norms influencing creativity; (2) a novel target domain for the study of creativity; (3) and a vast repository of empirical data which may permit that study itself to take novel forms. Not only is FOSS highly amenable to empirical analysis, while maintaining high ecological validity, but such investigations also have great potential for informing unresolved issues in the nature of creativity, with implications for innovation more broadly. The empirical study of FOSS, documenting the varied and continual input by different individuals (and the cognitive processes associated with such contributions), has the potential similarly to blur the venerable but arguably oversimplified distinction between ideation and elaboration and to shift the research dynamic toward a richer and more integrated perspective of creativity.

Acknowledgments This material is based upon work supported by the U.S. National Science Foundation under Grant No. 0855861. Our colleague Samir Chopra greatly influenced the development of many of the ideas discussed herein.

References

Amabile TM (1983) The social psychology of creativity. Springer, New York

Antoniades I (2007) Design of retrieval system. http://flossmetrics.org/sections/deliverables/docs/deliverables/WP2/D2.1-Retrieval_System_Design.pdf. Accessed 15 June 2012

Baer J (2011) Why grand theories of creativity distort, distract, and disappoint. Int J Creativity Probl Solving 21:73–100

Barnes N (2012) Science code manifesto. http://sciencecodemanifesto.org/. Accessed 23 May 2012

Benkler Yochai (2006) The wealth of networks: how social production transforms markets and freedom. Yale University Press, New Haven

Capra E, Francalanci C, Merlo F, Rossi Lamastra C (2011) Firms' involvement in open source projects: a trade-off between software quality and success. J Syst Softw 84(1):144–161
Chidamber SR, Kemerer CF (1994) A metrics suite for object oriented design. IEEE Trans Software Eng 20(6):476–493
Chopra S, Dexter S (2007) Decoding liberation: the promise of free and open source software. Routledge, New York
De Dreu C, Nijstad B, van Knippenberg D (2008) Motivated information processing in group judgment and decision making. Pers Soc Psychol Rev 12:22–49
DiBona C, Ockman S, Stone M (eds) (1999) Open sources: voices from the open source revolution. O'Reilly, Sebastopol
Dibona C, Cooper D, Stone M (eds) (2005) Open sources 2.0. O'Reilly, Sebastopol, California
Doctorow C (2008) Content: selected essays on technology, creativity, copyright, and the future of the future. Tachyon Publications, San Francisco
Ericsson K, Simon H (1984) Protocol analysis: verbal reports as data. MIT Press, Cambridge
European Commission (2006) Information Society Technologies (IST) 2005-6 Work Programme. http://cordis.europa.eu/ist/workprogramme/fp6_workprogramme.htm. Accessed 16 June 2012
European Commission (2010) Communication from the commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of Regions 'Towards interoperability for European public services' (Annex 2). http://ec.europa.eu/isa/documents/isa_annex_ii_eif_en.pdf. Accessed 16 June 2012
Eysenck HJ (1993) Creativity and personality: suggestions for a theory. Psychol Inq 4(3):147–178
Fayena-Tawil F, Kozbelt A, Sitaras L (2011) Think global, act local: a protocol analysis comparison of artists' and non-artists' cognitions, metacognitions, and evaluations while drawing. Psychol Aesthet Creativity Arts 5:135–145
Feller J, Fitzgerald B, Hissam S, Lakhani K (eds) (2005) Perspectives on free and open source software. MIT Press, Cambridge
FLOSSMetrics Consortium (2010) FLOSSMetrics final report. http://flossmetrics.org/docs/fm3-final-report_en.pdf. Accessed 3 June 2012
Free Software Foundation (2012) The free software definition. http://www.gnu.org/philosophy/free-sw.html. Accessed 27 January 2013
Getzels J, Csikszentmihalyi M (1976) The creative vision: a longitudinal study of problem finding in art. Wiley, New York
Ghosh R, Prakash V (2000) The orbiten free software survey. First Monday 5(7). http://firstmonday.org/issues/issue5_7/ghosh/index.html. Accessed 27 May 2012
Govindarajan V, Trimble C (2010) The other side of innovation: solving the execution challenge. Harvard Business Review Press, Cambridge
Guilford J (1950) Creativity. Am Psychol 5:444–454
Hutchins E (1995) Cognition in the wild. MIT Press, Cambridge
Jaarsveld S, van Leeuwen C (2005) Sketches from a design process: creative cognition inferred from intermediate products. Cognitive Sci 29:79–101
Kaufman J (2009) Creativity 101. Springer, New York
Kim KH, Cramond B, Van Tassel-Baska J (2011) Creativity and intelligence. In: Kaufman JC, Sternberg RJ (eds) The Cambridge handbook of creativity. Cambridge University Press, New York, pp 395–412
Koch S, Dueñas S, Herraiz I, Stamelos I, Samoladas I, Sowe S, Haaland K, Ghosh R (2009) High level studies. http://flossmetrics.org/sections/deliverables/docs/deliverables/WP5/D5.1-HighLevelStudies.pdf. Accessed 5 June 2012
Kounios J, Beeman M (2009) The aha moment! The cognitive neuroscience of insight. Cur Dir Psychol Sci 18:210–216
Kozbelt A (2006) Dynamic evaluation of Matisse's 1935 'Large Reclining Nude'. Empir Stud Arts 24:119–137
Kozbelt A (2008) Longitudinal hit ratios of classical composers: reconciling "Darwinian" and expertise acquisition perspectives on lifespan creativity. Psychol Aesthet Creativity Arts 2:221–235
Kozbelt A (2009) Ontogenetic heterochrony and the creative process in visual art: a précis. Psychol Aesthet Creativity Arts 3:35–37
Kozbelt A, Serafin J (2009) Dynamic evaluation of high- and low-creativity drawings by artist and non-artist raters. Creativity Res J 21:349–360

Kozbelt A, Dexter S, Dolese M, Seidel A (2012) The aesthetics of software code: a quantitative exploration. Psychol Aesthet Creativity Arts 6:57–65
Lakhani K, Wolf R (2005) Why hackers do what they do: understanding motivation and effort in free/open source software projects. In: Feller J, Fitzgerald B, Hissam S, Lakhani K (eds) Perspectives on free and open source software. MIT Press, Cambridge
Lammers S (1986/2006) Programmers at work. Microsoft Press, Redmond
Langley P, Simon H, Bradshaw G, Zytkow J (1987) Scientific discovery: computational explorations of the creative process. MIT Press, Cambridge
Lerner J, Tirole J (2005) The scope of open source licensing. J Law Econ Organ 21(1):20–56
Lessig L (2004) Free culture: how big media uses technology and the law to lock down culture and control creativity. The Penguin Press, New York
Lin Y (2006) Hybrid innovation: how OSS firms collaborate with the FLOSS community. Knowl Technol Policy 18(4):86–100
Martindale C (1990) The clockwork muse. Basic Books, New York
Metcalfe J, Wiebe D (1987) Intuition in insight and noninsight problem solving. Mem Cognit 15:238–246
Mockus A, Fielding R, Herbsleb J (2005) Two case studies of open source software development: apache and Mozilla. In: Feller J, Fitzgerald B, Hissam S, Lakhani K (eds) Perspectives on free and open source software. MIT Press, Cambridge, pp 163–209
Oram A, Wilson G (eds) (2007) Beautiful code. O'Reilly, Sebastopol
Raymond ES (2001) The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary. O'Reilly, Sebastopol
Rosen L (2004) Open source licensing: Software freedom and intellectual property law. Prentice Hall, Upper Saddle Hook
Sawyer R (2006) Explaining creativity: the science of human innovation. Oxford University Press, New York
Serafin J, Kozbelt A, Seidel A, Dolese M (2011) Dynamic evaluation of high- and low-creativity drawings by artist and non-artist raters: replication and methodological extension. Psychol Aesthet Creativity Arts 5:350–359
Shirky C (2010) Cognitive surplus: creativity and generosity in a connected age. The Penguin Press, New York
Silvia P (2008) Discernment and creativity: how well can people identify their most creative ideas? Psychol Aesthet Creativity Arts 2:139–146
Silvia PJ, Kaufman JC (2011) Creativity and mental illness. In: Kaufman JC, Sternberg RJ (eds) The Cambridge handbook of creativity. Cambridge University Press, New York, pp 381–394
Simonton D (1984) Creative productivity and age: a mathematical model based on a two-step cognitive process. Dev Rev 4:77–111
Simonton D (1999) Origins of genius: darwinian perspectives on creativity. Oxford University Press, New York
Sonnentag S, Niessen C, Volmer J (2006) Expertise in software design. In: Ericsson K, Charness N, Feltovich P, Hoffman R (eds) The Cambridge handbook of expertise and expert performance. Cambridge University Press, New York, pp 373–387
von Hippel E (2006) Democratizing innovation. MIT Press, Cambridge
von Krogh G, Haefliger S, Spaeth S, Wallin M (2012) Carrots and rainbows: motivation and social practice in open source software development. MIS Quart 26(2):649–676
Ward T, Smith S, Finke R (1999) Creative cognition. In: Sternberg R (ed) Handbook of creativity. Cambridge University Press, New York, pp 189–212
Weber S (2004) The success of open source. Harvard University Press, Cambridge