

34 SOFTWARE INNOVATION AS MAINTENANCE: Theorizing the Social in Software Maintenance

Allen Higgins

University College Dublin
Dublin, Ireland

Abstract

This paper entertains the notion that software maintenance and innovation are more closely related than is commonly accepted. We consider perspectives where innovation projects are understood as attempts to engineer both the social and the technological, where processes of innovation imply the configuring of users, communities, and artifacts through the work of maintenance, manifest perhaps as bricolage or drift. If this alternate interpretation of innovation is accepted, it implies a renewed sensitivity to research and method aligned to innovation settings, emphasizing subjects' interpretations, language, perceptions, behavior and even culture. This has implications for developing a deeper and more intimate understanding of processes surrounding software development.

Keywords

Software development, maintenance, programming, information systems development, diffusion of innovation

1 INTRODUCTION

Software undergoing maintenance is often regarded as a “mutable immutable,” where the idiom of maintenance is employed even though software does not wear out or degrade (Swanson 1976). Maintenance work is difficult and messy; patches must satisfy new demands without breaking existing installations and work as before (only better). It is better for your career to work on next generation technology, rather than being stuck on bug fixing or maintaining old versions within the straight jacket constraints of compatibility and legacy codebases (Riain 2000). Maintenance jobs are outsourced to low

Please use the following format when citing this chapter:

Higgins, A., 2007, in IFIP International Federation for Information Processing, Volume 235, Organizational Dynamics of Technology-Based Innovation: Diversifying the Research Agenda, eds. McMaster, T., Wastell, D., Ferneley, E., and DeGross, J. (Boston: Springer), pp. 475-479.

cost locations or shunted into the background noise of the workplace, and so the work of maintaining a venerable old version is often shunned by developers as they jockey for assignment to new product projects.

An alternate perspective, however, revalues the commonplace observation that software maintenance activities comprise merely the thick dynamic between customers and developers. E-mail and bug reports negotiate complaints, desires, trade-offs, insights, and solutions. These aspects of maintenance work, performed day-to-day on software still in development, suggest that the *idiom* of maintenance (connoting breakage, wear, and degradation over time) is not commensurate with the *activity* of maintenance (the co-production and mutual configuration of both users and objects).¹ What then if we adopt this alternate stance, that the work of software maintenance is instead a rich and messy process of innovation? This paper briefly explores the implications of conflating maintenance with innovation (suitably defined), and sketches a need for research attending to the situation and performance of maintenance work, to programming and configuring software as collective and multi-sited activities.

2 INNOVATION

Innovation is defined as the *bringing in or introduction of change* but the burgeoning innovation literature is largely concerned with the strategic production of novelty, usually conceived as new technology under the constraints of structural or individualist factors (Amabile et al. 1996; Christensen 1997; Schilling 2005; Trott 2005). Formal accounts of the organized production of software under the banner of innovation are likewise presented as processes of planned, purposeful engineering of novel products (Schilling 2005; Trott 2005). Software development is conceived as new product development (NPD), and the subsequent success (or otherwise) of technological innovation is considered to be path dependent; determined by factors like timing, market preparedness, technological superiority, or organization strategy. Key figures such as the engineer or architect, although central to these conceptions of invention and production, are rarely seen to be actively engaged with distant others in constructing the social contexts for their technological innovations. However, an alternate literature bedded in sociology presents another view, where so called engineer-sociologists are themselves engaged in building both the technologies and the social networks intended to take up an innovation (Callon 1987). This hints at another interpretation of innovation, revaluing the notion of *introduction* or *bringing in*, and shifting us toward a deeper concern with the use, appropriation, and adoption of changed technology by users or consumers. From this perspective we anticipate that innovation narrowly conceived under the rubric of NPD will lead to partial accounts, ignoring or dealing superficially with concepts of social power, interests, and action implicit in the sociality of *introduction*.

¹This position is informed by the author's on-going study in a software development company. Although new product development (NPD) is the high profile, strategic activity in the company, the metaphor of maintenance is found to be a dominant idiom within the workplace. The maintenance idiom is shared and employed by developers, testers, users, and others to describe, interpret, and explain much of their daily work.

Let us entertain this alternate paradigm, of software innovation as maintenance, where the idea of *maintenance* carries a stronger sense of *bringing in* or *introduction*. In this case introduction patterned on the notion of *xenia* (the Greek concept of hospitality or host-guest relations employed by Ciborra (2002)), transforms the ideation of innovation into a process of *hosting* technology within the social. Bringing in an innovation lets us blur the boundary between the host and the stranger (technology cast as the guest), as each accommodates the other if the materiality of software remains open-ended. What are the implications if maintenance work surrounding software is revealed as a valid location for processes and activities constituting innovation, and where might this understanding be positioned within the information systems literature?

3 PROCESSUAL INNOVATION

The concept of *processual* innovation (Brown and Eisenhardt 1995; Slappendel 1996; Von Hippel 2005) enriches an understanding of this setting, accounting for the generation of meaning by actors both using and developing software, through its maintenance and production. Slappendel (1996) reviewed the innovation literature and its underlying theoretical perspectives. Earlier studies implied two main levels of analysis (individualist and structuralist), but she also noted the growing acceptance of a third approach she termed the *interactive process* perspective. An integrative process view of innovation is an attempt to overcome the dialectic between individualist and structural perspectives, resolving the tension of their commensurability through an interactive dynamic. Von Hippel (2005), tracing the history of selected technology innovations, arrived at the pragmatic realization that products continue to be developed even when they leave the confines of a laboratory or engineering shop. He develops the concepts of *lead user* and *innovation communities* and concludes that innovation is a process of coproduction shared between the producer and the consumer of a new product. In a similar vein Ciborra (2002) discredits the appeal of new product development as the source of innovation, proposing instead to celebrate the situated activity of tinkering (or bricolage) as an incremental route to producing the technologies with which we work. Processes like *dérive* or *technology drift* present a “phenomenological middle ground, where the intentions of humans and non-humans mingle and blur; where learning and recalcitrance, hacking and inertia show up simultaneously” (Ciborra 2002, p. 84). Drift is a process of development and maintenance, a concept that links technology developed in the laboratory and the technology installed in the users’ environments and those in between: customers, support engineers, and programmers writing code. “[D]rift is about situated technology.... It is about technology in use, as experienced and seen from the swamp of contingent situations and practices, and not from the crisp, crystal-clear high ground of method” (Ciborra 2002, p. 90).

Processual perspectives on innovation, therefore, attend naturally to the diversity of actors interacting to stabilize their objectives through dynamics of crises, negotiation, and agreement. These strands of the IS literature claim that innovations are never fully designed top-down nor introduced in one shot, instead they are tried out through prototyping and tinkering. Innovation as *maintenance* would then be a collective and intrinsically social phenomenon resulting from the fluidity of software undergoing cycles

of packing and unpacking (construction, deconstruction, and reconstruction). Maintenance as bricolage, drift, and xenia offers a language for the production of innovation as *coproduction* (Grint and Woolgar 1997) involving the many actors participating in slow processes of transformation. We might think of ourselves then as hosts to the “gift” of software as it enters into the workplace, where it is shaped by, and in turn shapes, the lived phenomena of our work.

4 IMPLICATIONS FOR STUDIES OF SOFTWARE DEVELOPMENT WORK

A sketch of future research attending to the practices of maintaining software (software still in development) implies the necessity of observing those involved, as they interact, while maintaining and recreating software in use. It entails recording, tracing, and interpreting episodes of communication, dialectical conflict, and control (overt or pervasive), giving rise to adapted objects and conceptions of usage in maintenance settings. Field work patterned on ethnographic methods (Garfinkel 1967; Geertz 1973; Marcus 1998) provides a methodological approach to studying the domain through multi-sited ethnography (Marcus 1995) encompassing involved, detailed, longitudinal studies which privilege intersubjectivity generated in the contested environment of software maintenance.

References

- Amabile, T. M., Conti, R., Coon, H., Lazenby, J., and Herron, M. “Assessing the Work Environment for Creativity,” *Academy of Management Journal* (39:5), 1996, pp. 1154-1184.
- Brown, S. L., and Eisenhardt, K. M. “Product Development: Past Research, Present Findings, and Future Directions,” *Academy of Management. The Academy of Management Review* (20:2), 1995, pp. 343-378.
- Callon, M. “Society in the Making: The Study of Technology as a Tool for Sociological Analysis,” in W. E. Bijker, T. P. Hughes, and T. Pinch (eds.), *The Social Construction of Technological Systems*, Cambridge, MA: MIT Press, 1987, pp 83-103.
- Christensen, C. M. *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*, Boston: Harvard Business School Press, 1997.
- Ciborra, C. *The Labyrinths of Information: Challenging the Wisdom of Systems*, Oxford, UK: Oxford University Press, 2002.
- Garfinkel, H. *Studies in Ethnomethodology*, Englewood Cliffs, NJ: Prentice-Hall, 1967.
- Geertz, C. *The Interpretation of Cultures*, London: Fontana, 1973.
- Grint, K., and Woolgar, S. *The Machine at Work: Technology, Work and Organization*, Cambridge, UK: Polity Press, 1997.
- Marcus, G. E. “Ethnography in/of the World System: The Emergence of Multi-sited Ethnography,” *Annual Review of Anthropology* (24), 1998, pp. 95-117.
- Marcus, G. E. *Ethnography Through Thick and Thin*, Princeton, NJ: Princeton University Press, 1995.
- Riain, S. Ó. “Working for a Living: Irish Software Developers in the Global Workplace,” in M. Burawoy (ed.), *Global Ethnography: Forces, Connections, and Imaginations in a Postmodern World*, Berkeley, CA: University of California Press, 2000.

- Schilling, M. A. *Strategic Management of Technological Innovation*, New York: McGraw-Hill Irwin, 2005.
- Slappendel, C. "Perspectives on Innovation in Organizations," *Organization Studies* (17:1), 1996, pp. 107-129.
- Swanson, E. B. "The Dimensions of Maintenance," in *Proceedings of the Second international Conference on Software Engineering*, Los Alamitos, CA: IEEE Computer Society Press, 1976, pp. 492-497.
- Trott, P. *Innovation Management and New Product Development*, Upper Saddle River, NJ: Financial Times Prentice Hall, 2005.
- Von Hippel, E. *Democratizing Innovation*, Cambridge, MA: MIT Press, 2005.

About the Author

Allen Higgins is a researcher at University College Dublin and doctoral student at the University of Warwick under the supervision of Joe Nandhakumar. His studies are focused on the intersubjectivity of software development work, accessed ethnographically; seeking to describe aspects related to how programmers and users understand and access the objects they create; and processes of creating and maintaining the ensembles of actors composing processes of innovation. He can be reached by e-mail at allen.higgins@ucd.ie.