Creative and Open Software Engineering Practices and Tools in Maker Community Projects

Konstantinos Chorianopoulos
Ionian University
Plateia Tsirigoti 7, Corfu
49100, Greece
choko@ionio.gr
and NTNU

Letizia Jaccheri
Norwegian University of Science and Technology
Sem Sælands vei 7-9
Trondheim 7491, Norway
letizia@idi.ntnu.no

Alexander Salveson Nossum
Norwegian University of Science and Technology
Høgskoleringen 7A
7491 Trondheim, Norway
alexander.nossum@ntnu.no

ABSTRACT

Processing, Arduino and the growth of the associated communities of practice, also called Maker communities, has motivated a broader participation of non-technical users in the engineering of interactive systems. Besides online sharing, Maker communities meet regularly and share knowledge for various purposes (e.g., creative hacking, social networking, lifelong learning). In the context of Maker communities, the understanding of engineering interactive systems (e.g., motivations, objectives, collaboration, process, reports) and the design of the respective tools (e.g., end-user programming for artists, or children) are not well documented. As a remedy, we present a coherent overview of related work, as well as our own experiences in the organization and running of Maker workshops. The tutorial format (lecture and hands-on workshop) benefits both practitioners and researchers with an understanding of creative software tools and practices. Moreover, participants become familiar with the organization of Maker workshops as 1) a research method for understanding users, 2) an engineering process for interactive computer systems, and 3) a practice for teaching and learning.

ACM Classification Keywords: H.5.2 [User Interfaces]: Prototyping; Evaluation/methodology. D.2.2 [Software Engineering]: Design Tools and Techniques; User Interfaces.

Keywords

Open Source, Creative process, Maker, collaboration.

INTRODUCTION

Although the software engineering discipline has mostly evolved in the context of large-scale corporate projects, there have always been small-scale ad-hoc efforts by communities of hobbyists (e.g., shareware, independent games). The skills and motivations of Maker communities are rather broad. Nevertheless, the design of the most popular software tools has mostly regarded corporate users, or only those hobbyists with strong engineering backgrounds (e.g., VisualStudio, Eclipse, XCode). More recently, software developed by makers and for makers [1] has emerged as a new software category at the intersection between software and art. Artists need software technology for creating and evolving their artwork [2]. Technologists have long regarded the contact with artists as a source of inspiration for innovation [3]. Industry is also paying attention, because users have always been recognized as a source of innovation [4]. For example, an increasing number of companies are getting involved in OSS projects [5]. Tools for creativity have emerged as an important subject of study within computer science research [6]. Several popular rapid prototyping tools, such as Processing, Arduino, and Scratch [1] [7] have their roots in this intersection of open source engineering and creativity. In this tutorial, we suggest that the intrinsic motivation of hobbyists and Maker communities have many significant benefits for research, engineering, and learning.

OPEN SOURCE TOOLS

A technology is considered open source if its design (e.g., code, diagram, parts-list) is available to everybody for inspection, use, and modification. Notably, users of open source are not paying customers but potential co-developers. End-user development has considered some issues related with software and tools, but has not considered other emerging aspects of open source, such as community collaboration and open hardware. The most important characteristics and success factor of open source projects are associated with communities of users and of developers. The quality of collaboration in the community is crucial for any software project [8]. Members of each community are connected and assist each other via computer mediated communication tools (e.g., wikis, forums), as well as via real world meetings (e.g., Maker events, Hacker-spaces). In particular, Maker events have built upon the open source approach by offering a localized social gathering, which reinforces community as well as knowledge sharing. Moreover, Maker communities produce tangible interactive systems with the help of affordable fabrication methods (e.g., 3D printers) and tools (e.g., Arduino), thus they have novel needs in terms of tool chain support.

EICS'12, June 25–26, 2012, Copenhagen, Denmark.
ACM 978-1-4503-1168-7/12/06.

CREATIVE AND COLLABORATIVE PROCESS

The inner joy of creation and unselfish cooperation has often been identified as an important asset of the open source developer culture, bringing it, according to Castells [9] close to the world of art. For Castells, the Internet not only serves as a means for distribution of final blueprints, but also serves as a shared platform for a process that aims to create new blueprints. Innovative and creative businesses are often found within the field of computer science. These are often faced with similar collaboration issues, for instance when one or more stakeholders to a project have different backgrounds than the rest of a team. Or when the task is to "think outside the box" and develop creative solutions. The social issues experienced in these settings may be an obstacle that results in sub-optimal solutions [10].

Collaboration between several, co-located or distant, persons can be a complex task. This is an issue shared with most intellectual activities. In previous creativity work [11] [12], this issue has been identified, when computer engineering practitioners are working together with non-engineers or other branches of engineering. Applying and adapting methods from the field of social psychology have proven to be successful in optimizing the collaboration in heterogeneous groups. These methods approach the issues in a social manner and generally aim at optimizing the social issues and in effect optimizing the end results of the collaboration.

LEARNING OBJECTIVES AND STRUCTURE

Open source tools for creativity: In this tutorial, we are leveraging previous empirical evidence [13] and we chose to focus on open tools and creative processes. The format has also been motivated by significant related work in open source, such as Processing, Arduino, and Scratch.

Community documentation and reporting: The proposed workshop (hands-on) format employs several techniques for documentation and reporting. The ArTe blog links to a set of reusable resources, among which three master level and a PhD course in the interdisciplinary field of software and art. The blog also documents a set of creative workshops that have been offered by our research group during the last years. This documentation includes pictures, videos, and code, which can be reused and reflected upon.

Research issues: The tutorial regards several research questions that are intended to be explored and to be a basis for further elaboration:

- "How can we support Maker communities with updated practices and tools?"
- "How can the creative process of Maker communities contribute to the existing engineering theory of interactive systems?"

Guidelines for organizing and hosting your own Maker workshop: The tutorial includes experience based creativity sessions, which are based on a process that has been documented and validated through interviews with participants and analysis of the collected data and developed artifacts [13]. Thus, participants will become empowered with methods that enhance their research, engineering, and teaching practice.

REFERENCES

1. Noble, J. *Programming Interactivity: A Designer's Guide to Processing, Arduino, and Openframeworks*. O'Reilly Media, 2009.
2. Trifonova, A., Jaccheri, L., and Bergaust, K. Software Engineering Issues in Interactive Installation Art. *International Journal on Arts and Technology (IJART)* 1, 1 (2008), 43-65.
3. Harris, C. Art and innovation: the Xerox PARC Artist-in-Residence program. MIT Press, Cambridge, MA, USA, 1999.
4. E von Hippel. 1986. Lead users: a source of novel product concepts. Management Science 32, 7 (July 1986), 791-805.
5. Herbsleb. J.D. Global Software Engineering: The Future of Socio-technical Coordination. In *Future of Software Engineering (FOSE '07)*. IEEE Computer Society, Washington, DC, USA, 188-198.
6. Shneiderman, B. Creativity Support Tools Accelerating Discovery and Innovation. Communication of the ACM 50, 12 (2007), 20-32.
7. Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E. The Scratch Programming Language and Environment. Trans. Comput. Educ. 10, 4, Article 16 (November 2010), 15 pages.
8. Dittrich, Y., Randall, D. W., Singer, J. Software Engineering as Cooperative Work. *Comput. Supported Coop. Work* 18, 5-6 (December 2009), 393-399.
9. Castells, M. Rise of The Network Society (Castells, Manuel. Information Age, 1.) (Vol 1). Wiley, 1996.
10. Johnson, D.W. and Johnson, F.P. Joining Together: Group Theory and Group Skills (10th Edition). Pearson, 2008.
11. Jaccheri, L., Sindre, G. Software Engineering Students meet Interdisciplinary Project work and Art. 11th International Conference Information Visualization, IV '07, IEEE Computer Society (2007), 925-934.
12. Chorianopoulos, K., Rieniets, T. City of collision: an interactive video installation to inform and engage. In IET Conference Publications, 2007, 502-509.
13. Høiseth, M. and Jaccheri, L. Art and Technology for Young Creators. Entertainment Computing - ICEC 2011 - 10th International Conference, Springer (2011), 210-221.