

Creative and Open Software Engineering Practices and Tools in Community Projects

Konstantinos

Chorianopoulos

Department of Informatics

Ionian University

Plateia Tsirigoti 7, Corfu

49100, Greece

choko@ionio.gr

Letizia Jaccheri

Department of Computer and

Information Science

Norwegian University of

Science and Technology

Sem Sælands vei 7-9

Trondheim 7491, Norway

letizia@idi.ntnu.no

Alexander Salveson

Nossum

Division of Geomatics

Norwegian University of

Science and Technology

Høgskoleringen 7A

7491 Trondheim, Norway

alexander.nossum@ntnu.no

ABSTRACT

In this tutorial, we explore software engineering practices and tools that are suitable for a growing number of amateur creators and artists. Although the software engineering discipline has mostly evolved in the context of large-scale corporate projects and clients, there have always been small-scale ad-hoc efforts by communities of hobbyists (e.g., shareware, indy-games). As a matter of fact, 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, the introduction of open source tools such as Processing and Arduino has motivated a broader participation of non-technical users in the engineering of interactive systems. Maker communities meet regularly and share resources and knowledge for various purposes (e.g., creative hacking, fun, social networking). In the context of maker communities, the understanding of the software creation practices (e.g., collaboration, sharing) 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.

Keywords

Open Source Software, Art, Creative processes

INTRODUCTION

Software developed by artists and for artists is emerging as new category of software and software dependent art projects are growing in numbers, drawing the attention of software developers and artists [16]. Artists need software technology for creating and evolving their artwork [15]. Theoreticians aim at understanding the consequences of software to art practices [4]. Technologists see the contact with artists as a source for innovation [5]. Computer science researchers recognize the importance of new media art as a legitimate research field. Tools for creativity are a subject of study within computer science research [11] [14]. ‘New media artists realize their desire for personal expression

with powerful development environments that support animation, music, or video editing tools’ [14]. Open source tools for creativity, such as Processing [12], Arduino [2], and Scratch [13] have their roots in this intersection of open source software and creativity. These two fields of study and practice are important for the work presented in this tutorial.

OPEN SOURCE SOFTWARE

A software system is open source if its code is available to everybody for inspection, use, and modification. Users of OSS are not paying customers but potential software co-developers. OSS was born as a movement based on contributions of volunteers. However, an increasing number of companies are getting involved in OSS projects [17].

Use and further release of modified version of an OSS system are regulated by a license. OSS is much more than the possibility to change the code. The most important characteristics and success factor of OSS projects are associated with communities of users and developers. The degree of activeness of the community is crucial for an OSS project. Each user of an active community is not isolated but part of a community. Members of each community are connected and help each other via mailing lists, forum and IRC channels.

CREATIVITY AND COLLABORATION

The inner joy of creation and unselfish cooperation has often been identified as an important asset of the OSS developer culture, bringing it, according to Castells [3] close to the world of art. Castells anticipates art as a growing area of the Internet, stating that ‘open source art is the new frontier of artistic creation’. For Castells, the Internet not only serves as a means for distribution of artifacts, but also serves as a shared platform for a process that aims to create new artistic artifacts.

Collaboration between several, distant, persons can be a complex task. This is an issue shared with most intellectual activities. In previous work [7] this issue has been identified, when computer engineering students 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. This has proven successful when applied on students with a multi-disciplinary background working with innovative, new media, solutions [1].

Innovative and creative businesses are often found within the field of computer science. These are often faced with similar 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 [9].

DESCRIPTION AND INSPIRATIONAL QUESTIONS

In this tutorial [6,8], we chose to focus on OSS and its intersection with art and creative processes. This choice is motivated by significant related work in OSS and art and available OSS for art; like Processing [10,12], Arduino [2], and Scratch [13].

Pictures will be used as a mean to record and reflect about the creative activities. The ArTe blog [18] links to a set of reusable resources, among which three master level and a PhD course in the shared field of software and art. The blog also documents a set of creative workshops that have been offered by the same teacher and her group during the last years. This documentation includes pictures, videos, and code fragments to be reused.

The tutorial is motivated by several inspirational research questions that are intended to be explored and be a basis for reflection. Examples of questions are:

- RQ1 “How can creative processes aid in professional software engineering?”
- RQ2 “How can an understanding of OSS communities as an art form enhance open software engineering practices and tools?”
- RQ3 “How can new media, location based services, ubiquitous computing become a form of art?”

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 [6].

REFERENCES

1. Aasbakken, M. *Evaluating User Experiences from Interactive Art Installation, project report*. 2011, available at www.artentnu.com
2. Arduino. Arduino Web Site. 2011. <http://arduino.cc>.
3. Castells, M. *Rise of The Network Society (Castells, Manuel. Information Age, 1.) (Vol 1)*. Wiley, 1996.
4. Halonen, K. Open Source and New Media Artists. *Human Technology - An Interdisciplinary Journal Humans in ICT Environments* 3, (2007), 98-114.
5. Harris, C. *Art and innovation: the Xerox PARC Artist-in-Residence program*. MIT Press, Cambridge, MA, USA, 1999.
6. Høiseth, M. and Jaccheri, L. Art and Technology for Young Creators. *Entertainment Computing - ICEC 2011 - 10th International Conference*, Springer (2011), 210-221.
7. Jaccheri, L. and Sindre, G. Software Engineering Students meet Interdisciplinary Project work and Art. IV, IEEE Computer Society (2007), 925-934.
8. Jaccheri, L. Open Software and Art: A Tutorial. *Entertainment Computing - ICEC 2011 - 10th International Conference*, Springer (2011), 468-471.
9. Johnson, D.W. and Johnson, F.P. *Joining Together: Group Theory and Group Skills (10th Edition)*. Pearson, 2008.
10. Noble, J. *Programming Interactivity: A Designer's Guide to Processing, Arduino, and Openframeworks*. O'Reilly Media, 2009.
11. Oates, B.J. New Frontiers for Information Systems Research: Computer Art as an Information System. *European Journal of Information Systems* 15, (2006), 617-626.
12. Processing. Processing Web Site. 2011. <http://processing.org>.
13. Scratch. Scratch Web Site. 2011. <http://scratch.mit.edu/>
14. Shneiderman, B. Creativity Support Tools Accelerating Discovery and Innovation. *Communication of the ACM* 50, 12 (2007), 20-32.
15. 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.
16. Ahmed, S.A. , "Extending Software Engineering Collaboration towards the Intersection of Software and Art", PhD thesis, NTNU 2011:77, ISBN 978-82-471-2684-4 (electronic), ISBN 978-82-471-2683-7 (printed), ISSN 1503-8181. Article-thesis: 239 pages available at <http://www.idi.ntnu.no/grupper/su/publ/phd/salah-phd2011.pdf>
17. Herbsleb, J.D., “Global Software Engineering: The Future of Socio-technical Coordination,” 29th International Conference on Software Engineering- Future of Software Engineering (ICSE), 2007, pp. 188-198.
18. ArTe Blog. Jaccheri, L. et al.. 2011. www.artentnu.com/