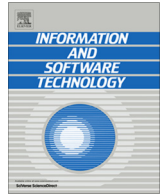




Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Understanding reuse of software examples: A case study of prejudice in a community of practice

Ohad Barzilay^{a,*}, Cathy Urquhart^b

^a Tel Aviv University, Israel

^b Manchester Metropolitan University, Manchester, UK

ARTICLE INFO

Article history:

Received 7 April 2013

Received in revised form 25 February 2014

Accepted 26 February 2014

Available online xxx

Keywords:

LinkedIn

Qualitative research

Grounded theory

Virtual focus group

Human aspects

Open source development

ABSTRACT

Context: The context of this research is software developers' perceptions about the use of code examples in professional software development.

Objective: The primary objective of this paper is to identify the human factors that dominate example usage among professional software developers, and to provide a theory that explains these factors.

Method: To achieve this goal, we analyzed the perceptions of professional software developers as manifested on LinkedIn online community. We analyzed the data qualitatively using adapted grounded theory research procedures.

Results: The research yields an initial framework of key factors that dominate professional developers' perception regarding example usage. We use the theoretical lens of *prejudice theory* to put these factors in a broader context, and outline initial recommendations to address these factors in professional organizational context.

Conclusion: The results of this work, in particular the use of qualitative techniques – allowed us to obtain rich insight into key human factors that affect professional software developers, and enrich the body of literature on the issues of reuse. These factors need to be taken into account as part of an organizational reuse strategy.

© 2014 Published by Elsevier B.V.

1. Introduction

This paper examines an important issue for software developers – that of software reuse, and how this might be perceived in relation to code duplication. In the software engineering community, *code duplication* has been widely acknowledged as a bad practice [57]. When copying and pasting code snippets found elsewhere in the code base, the developer misses opportunities to abstract the functionality in question, and in turn, reduces code quality. In addition, redundant code often results in consistency issues, which also affect the quality of the overall system [50].

Code reuse, on the other hand, is widely accepted as a good practice. The literature advocates the reuse of existing code, and mentions several benefits of it including: increased productivity [89], improvement of code quality [68] enforcement of design

consistency [31,77] and of coding standards [17], and the establishment of an effective knowledge transfer mechanism both within and outside the organization [92].

In some cases, however, code reuse may be perceived by software developers as code duplication. Given the recent massive availability of code online, in open source projects, technical blogs, and Q&A websites [10,97], this issue is worthy of investigation. On the one hand, incorporating this online code in production may be considered as a modern manifestation of code reuse – already written code that can spare the software developer the effort to “re-invent the wheel”. On the other hand, using this code involves ‘copying and pasting’ it, an action that on itself, serves for many developers as a warning sign, an indicator, of the banned code duplication activity.

In this paper, we focus on developers' perception of (re)using code examples – existing code snippets that are used in a new context. Our definition of a code ‘example’ is broad; some of the code examples which appear on the Internet were not written in order to be reused. Code examples may accompany answers on Q&A sites [12], illustrate and idea in an online tutorial, or even be extracted from an open source project [98].

* Corresponding author. Address: Tel Aviv University, Recanati Building, Ramat Aviv, Tel Aviv 69978, Israel. Tel.: +972 3 6406292; fax: +972 3 6409983.

E-mail addresses: ohadbr@tau.ac.il (O. Barzilay), c.urquhart@mmu.ac.uk (C. Urquhart).

Research questions:

- How do software developers perceive example usage in their work, and in particular a usage that involves copy and paste? What arguments do they use to justify their position? Which concerns do they consider?
- How can these perceptions be explained?

In this study, a community of software developers on LinkedIn was asked to describe its practices when using code examples. Developers' responses revealed a variety of perspectives on using code examples. We outline those different perspectives by dividing these developers into three groups according to their example usage characteristics: those who use examples habitually, those who avoid using examples, and those who make limited use of examples. Further analysis of the findings using grounded theory [102] methods revealed nine human factors that dominate developers' perceptions. It is suggested that developers' approach is dominated by their personality, and affected by concerns such as their community identity, ownership and trust. We find that developers' perception of such reuse goes beyond the activities and practices, and that some developers associate the use of code examples with negative character. Some of these developers stereotype habitual example users as inferior and unprofessional.

In this paper we use the *prejudice perspective* [86,23] in order to examine how certain software developers are 'prejudiced' against the use of code examples, and consequently against developers who do not share their views on this matter. The term prejudice is of course somewhat loaded; in the social sciences it is commonly used in the context of stereotyping, and discrimination against certain groups, which results in racism, nationalism and sexism. While we would not claim that in the context of software development the implications of prejudice against use of examples are as severe, we do find this perspective supremely useful to examine the adoption and avoidance of using certain professional practices, for reasons which will be explained in more depth later.

It should be noted that not only human aspects are associated with example usage – there are some other issues involved in this activity such as engineering aspects (e.g. search techniques and tools) and legal issues (e.g. copyright and licensing). These issues are outside of the scope of our discussion; however, we believe that these challenges can be mitigated with proper training and organizational support (e.g. teaching developers which code could they use, and on what circumstances). This approach is also supported by Rotenberger et al. [91] and Morisio et al. [75].

This work is built upon previous work of Barzilay et al. [11], using social media to study the diversity of developers' perception regarding example usage. This paper seeks to make the following contributions. First, by highlighting the concerns of software developers with respect to example usage, we identify additional reuse barriers and augment the existing body of knowledge in software reuse. Second, we demonstrate how human concerns dominate developers' behavior – in this respect, our case study of example usage can be used as a test bed to examine a larger array of activities and their associated human concerns. In addition, we extend prejudice theory to the domain of software engineering, and show that developers' perceptions regarding a professional activity extend beyond the activity itself, and that they attribute characteristics to other professionals that do not share their views. Our work also demonstrates how the views of an online community can be effectively surveyed and analyzed, and we share our methodology for doing so.

In the next section, we present our chosen theoretical lens of prejudice theory, and discuss relevant literature on software reuse. We then present our methodology for the study, which was carried out using a virtual focus group in a social networking community. In the fourth section, we explain our data analysis, and in the fifth

section present our findings using the lens of prejudice theory. Next, we integrate our findings regarding the human aspects involved with example usage with the literature. Finally, we discuss the limitations of this study, and draw some conclusions and implications about the reuse of software examples in software development communities.

2. Related work

In this section, we first discuss theories of prejudice and why they may be helpful for our research problem. We then discuss literature on software reuse.

2.1. Theories of prejudice

Prejudice literature is used in social sciences to study racism, sexism and discrimination. Why then might it be relevant to software example reuse? We noticed that the developers in our study had strong and entrenched opinions about reuse, and their beliefs are extended toward other software practitioners who hold counter opinions. Software construction is a human-intensive activity [108], as such, many aspects of it may be dealt from a human perspective. Using prejudice theory [86,6,23] we explain opposition to the use of code examples as *preconceived opinions originated in a narrow implicit context, in which they first encountered*. Chauvinism, for example, may result from growing in a culture in which women work only in certain jobs. Racism can build when a certain group is encountered only in a specific negative context. Even when behaviors are undeniably caused by situational factors, people will sometimes favor dispositional explanations – a misjudgment known as the “fundamental attribution error” [90]. The outline of this overview is adapted from Plous [86].

The majority of social scientists agree that “prejudice” involves a prejudgment, usually negative, about a group or its members [29,49,79], even if their precise definitions vary somewhat. We should also stress here that prejudice is not merely a statement of opinion or belief, but an attitude that can include feelings such as contempt, dislike, or loathing [29]. Below we outline some key dimensions of prejudice theory.

2.1.1. Authoritarian personality

Adorno et al. [3] concluded that the key to prejudice lay in what they call an “authoritarian personality.” They described authoritarians as rigid thinkers who obeyed authority, saw the world as black and white, and enforced strict adherence to social rules and hierarchies. Authoritarians harbor many double standards and hypocrisies, without realizing it [7]. Adult authoritarians travel in tight circles of like-minded people; they often think their views are commonly held in society, that they are the “Moral Majority” or the “Silent Majority” [7].

Authoritarian people are more likely than others to harbor prejudices against low-status groups [3,86]. Furthermore, Social Dominance Theory states [96] that society can be viewed as “*group-based hierarchies. In competition for scarce resources such as housing or employment, dominant groups create prejudiced “legitimizing myths” to provide moral and intellectual justification for their dominant position over other groups and validate their claim over the limited resources*” [96].

2.1.2. Categorical thinking

Allport [6] suggests that prejudice is partly an outgrowth of normal human functioning, and relates it to categorical thinking:

The human mind must think with the aid of categories...Once formed, categories are the basis for normal prejudgment. We cannot possibly avoid this process. Orderly living depends upon it.

[[6] p. 20]]

Download English Version:

<https://daneshyari.com/en/article/10366751>

Download Persian Version:

<https://daneshyari.com/article/10366751>

[Daneshyari.com](https://daneshyari.com)