

Foregrounding invisible work practices: the customization of generic software systems

Zeferino Saugene

Department of Informatics, University of Oslo
Ole-Johan Dahls hus, Gaustadalléen 23B
0373 OSLO
zeferino.saugene@gmail.com

Jens Kaasbøll

Department of Informatics, University of Oslo
Ole-Johan Dahls hus, Gaustadalléen 23B
0373 OSLO
jensj@ifi.uio.no

ABSTRACT

One of the consequences of globalization has been the increased use of generic and globally developed software systems. Projects involved in the development of such software systems, especially in low and middle income countries, emphasize the presence of multi-faceted challenges. This position paper examines work practices, stakeholders and actions that are part of the software customization stage of global software development. Empirical material was gathered by observing the articulation work performed by software customizers in real cooperative work settings. Findings demonstrate that customizers and core software developers have limited insight into each other's practices, that managers have too little knowledge about the customization process, the skills needed, and the informal arrangements to overcome problems.

Author Keywords

Global software development; generic software system; software customization; articulation work; invisible work practices; CSCW.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

With globalization, information technology (IT) industry has evolved to the point software artefacts are now developed through global software alliances (see Sahay [1]). One of the factors that make these alliances to function is the possibility of local teams to benefit from software systems produced by others. We refer to these software systems, targeting a wide range of users, as generic software systems (GSS).

Global software development (GSD) is becoming a common practice through which organizations are developing these GSS; and is characterized by stakeholders from different locations having different culture and background, collaborating in the development[2].

GSD is, primarily, performed by a team of *developers* cooperating globally [3] - those that initiate the development process by specifying the design patterns, components and frameworks, and defining the artefacts' default settings. Yet, one important step in the GSD is the software customization activity, which consists of mundane activities of task-artefact fit. In other words, software customization refers to a set of tasks performed to fit generic IT artefacts into specific local contexts, and keeping them working. Software developers also collect and validate requirements through customization. This activity is trusted to *customizers* - those responsible for taking a generic IT-artefact and tailoring it to suit the needs of a group of users [4].

Most literature on GSS reduce software customization to taking a generic framework from development settings, adjusting it to the use settings, testing and problem solving, and using the lessons (e.g., problems, new needs and demands) from the use settings to enhance and evolve the framework. Nonetheless, customization of GSS also concerns the means and practices that customizers adopt to articulate their cooperative activities. These means and practices are usually left in the background when software customization is described.

Computer Supported Cooperative Work (CSCW) scholars regard work that is left in the background as invisible work. Invisibility is described by Star and Strauss [5] as "the neglect of representing knowledge and skills as formal work"[6, pg. 275]. Drawing from a research performed in three GSS customization projects, we found that (1) most project managers do not know in detail what is needed to perform the customization of GSS; because of that (2) they don't pay attention to skills and experiences of the people they select to be customizers; (3) actions performed at the local settings are invisible to developers and these developers are also invisible to the local customizers; and finally (4) the actions performed to make customization successful, e.g., 'informal' channels used to get help with customization, are invisible or not explicit to the project managers.

To capture the invisible work involved in making technology work, scholars adopted the concept of articulation work [5, 7-9]. Articulation work is defined as “work that gets things back ‘on track’ in the face of the unexpected, and modifies action to accommodate unanticipated contingencies” [5, pg. 10]. Articulation work also includes the continuous effort required in order to bring together discontinuous elements – of organizations, of professional practices, of technologies – into working configurations [8]. In this way we think that software customization is an articulation work that can help to make visible actions that are usually invisible to GSD stakeholders.

RESEARCH FOCUS

In this research, we describe a work performed by teams that participated in the development of a GSS called District Health Information Software (DHIS). DHIS is a free and open source software framework designed to support health workers and managers at all administrative levels through a balance between flexibility and standardization, and with a strong emphasis on using information for local action. The framework was developed by a global south-south-north collaborative network called Health Information Systems Programme (HISP). HISP, from the University of Oslo in Norway, has been engaged in the process of strengthening Health Information Systems (HIS) in the South for more than a decade, and these efforts have touched more than thirty countries in Asia and Africa.

As HISP’s key component, DHIS has been designed and developed for use in these countries. Customization is one of the work practices adopted to institutionalize DHIS. From 2009 through 2013, we have been involved in several DHIS2 customization projects, to highlight (1) work performed by HISP Malawi to revamp their Health Information System through customization of DHIS; (2) work by OGUMANIHA¹ in Mozambique; and (3) work by the National Institute of Public Health in Guinea-Bissau which customized DHIS to integrate information produced by the different health programs. The goal of our involvement in the customization projects was to produce a better understanding of how a globally-distributed team using DHIS manage to, successfully, overcome the obstacles associated with GSD, as well as to contribute to the development of the GSS; that is, build understanding of the work involved in customizing and gathering requirements in GSD.

¹ a group of seven NGOs (World Vision, Adventist Development and Relief Agency, ACDI/VOCA, International Relief and Development, Friends in Global Health at the Vanderbilt University, John Hopkins University’s Centre for Communication Programs, and Red Cross Mozambique), that customized DHIS to manage data on 92 indicators implemented under the auspices of the Ministries of Health, Agriculture, and Public Works and Housing.

Findings show that, the institutionalization of DHIS is supported by an inter-disciplinary team of developers, users and customizers. Although members of this team have disparate perspectives and cultures, together they make a DHIS community. Formally, members of this community communicate through message feature embedded within the software itself, mailing lists and DHIS workshop academies. Around DHIS, people use shared artefacts such as the official website and manuals (developer, customizer and user), to improve the understanding of the processes and workflows.

Analysis was conducted on the core customization activities including software testing, adaptation and problem-solving. Not surprisingly, in the course of the project, the process of customization presented challenges to the local GSD teams. Because of that we were able to examine the practices adopted by customization teams to build capacity, and well as the practices adopted especially by software developers to make explicit work that was to some extent invisible to all GSD stakeholders. These are discussed in the next section.

REFLECTIONS

As software systems development moves from single to group tasks and communication, the jobs of tuning, adjusting, and monitoring use and users grows in complexity [5]. The projects investigated in this research helped us to identify several invisible work practices and their implication to the GSD. Four main categories of invisibility were observed, and are as follows:

(1) Actions performed at the local settings are invisible to developers and developers are also invisible to the local customizers

As opposed to custom software systems, GSS are tools designed and developed for general use, i.e., applied to a wide community [10]. While they provide several benefits a number of activities need to be performed at the local settings to make them fit the needs and demands. Because of that, software customization phase of GSD is considered as an situated activity, meaning that, it both shapes and is shaped by the society within which it occurs [11], and is performed “by engaging more deeply in the empirical details of organizational life on the ground” [12, pg. 24]. This study illustrated that most of these customization actions are not accessible to the developers until problems are faced by local customizers which require intervention of other members of GSD community.

Moreover, it is well known that freely available GSS such as DHIS may be adopted by project managers without having to contract the developers. This absence of contract or contact with developers implies that, local teams have to be created and will be responsible to inspect the software tool, adapt and be responsible for any problems that might pop-up during the customization activity. Often local customizers never meet, in person, members of developer

organizations. Examples from this study show OGUMANIHA who decided to adopt DHIS based on preliminary work conducted by their consultant. Before its team faced problems or wanted to enable spatial analysis features they were able to perform most of the customization work and did not see any reason for meeting the developers.

These problems somehow impede communication and coordination to occur. For example, associating language diversity to the fact that developers are not known it can be difficult for customizers to communicate with developers.

(2) Managers pay little attention to skills and experiences of the peoples they select to be customizers

Being developed through global alliances, GSS offer limitations to local managers with regard to opportunities for learning the experiences from other settings.

Moreover, Clement [13] argue that decisions to adopt GSS are “conventionally made by managers, often without consulting their support staff” [13, pg. 326]. Because of this and also the fact that healthcare organizations of low and middle income countries (LMIC) face problems of lack of skills, the problems of lack of financial resources to hire the most skilled people, the skills of local customizers are usually unknown by or invisible to project managers. The research has shown that becoming a customizer does not necessarily require proven customization experience or skills. The customizer projects we studied had among their staff people who had been hired for their technical expertise and other who were not. For instance, while software customizers in Malawi and Guinea-Bissau were elected by the project coordination team, in Mozambique they were selected based on a public tender. The latter had an extensive job description which included description of the tasks, and the tools which were going to be used. For the case of Guinea-Bissau customizers were selected based on their willingness to join the team. These problems have influenced the creation of the artefacts, as well as triggered articulation work to take place.

(3) Most project managers do not know in detail what is needed in order to perform the customization of GSS

Introducing complex technologies such as GSS into LMICs work settings means not only that staff have skills and competence but managers know what work is required. This is supported by [7, 9] when state that creation shared space require coordination practice, which in turn cannot happen unless they have knowledge of what is supposed to be done. The research illustrated that most managers are more concerned with the benefits that the GSS offer, leaving the work necessary to earn these benefits in the background.

In this research we observed that in some settings managers were only concerned with the result of the customization.

Issues such as capacity building toward sustainability of the system were taken-for-granted. Because of this coordination was little and it was done mainly by the developer organization. If it were not for the intervention of partners, for example, customizers would not have participated in DHIS academies and it would not have been able to have guest customizers in the settings to help them solve problems.

Due to the distributed nature of the GSD, actors involved in the customization of the technology perform articulation work to manage the consequences of this distribution[6]. Moreover, because most of the work is done by others in other stages of GSD, customization is usually considered as an activity that reduces the work and costs of production software systems. This conceptualization is problematic because the work involved in building capacity and communicating with the developers becomes hidden. Consequentially, project managers do not pay enough attention to the customization tasks, and as result they are surprised when articulation work issues come up.

(4) The actions performed to make customization successful, e.g., ‘informal’ channels used to get help with customization, are invisible or not explicit to the project managers

Findings from this research show that new adopters (end-users and customizers) may find themselves confronted with the highly complex and, at first glance, not very transparent GSS. This has made the activity of customization very challenging. Example of turbulences included DHIS not allowing mimicking business processes workflow, difficulties to share data elements between different health programs, and non-availability of offline data-entry and import/export features.

In response to these problems, various actions performed toward the effectiveness of customization activity, were identified. Special focus is paid to the roles of the different stakeholders in the process. For example, to address customization challenges, local GSD teams benefit from collaboration with the developers of the technology, as well as from customizers from other countries. In other words, the team benefit from informal customizer-customizer and customizer-developer interactions using emails, personal calls (e.g., via Skype) and chats.

It was observed that to quickly address challenges, customizers benefited from many channels including software artefacts, software developers and fellow customizers. Specifically, findings illustrated examples where local customizers were visited (face-to-face) by people involved in similar projects in other contexts referred as ‘guest’ customizers. The research illustrates also the use of DHIS2 academies as source of customization capacity. Most of these channels emerged during the customization processes; managers were not prepared financially to send their customizers to attend the

academies. In Malawi and Guinea-Bissau, customizers had to rely on financial support from donors.

However, the fact that the people were shared between interventions provided a common ground for mutual understanding between local and guest customizers. It was observed that during the tailored trainings, while local customizers learned how to perform the software customization activity, the 'guest' customizers gained knowledge about local particularities, as well as delivered to the local people the experiences from other settings. In other words, together the teams (local and guests) developed a common understanding on how the technology would be customized. This practice helped customizers to, at a given moment in time, collect information on a specific topic and proceed with the customization process. The practice allowed also customizers to benefit from experiences of 'guest' customizers. By obtaining quick responses from community, it helped them to keep the customization activities up to speed.

CONCLUSION

The process of GSD goes through several stages; one fundamental stage is software customization. Recognizing this as an essential GSD phase is very important in order to put GSS into use; however the invisible work of customizing, which may be systematically and sometimes deliberately ignored, might challenge the overall GSD process.

This position paper examined software customization activities of three instances of a GSD project called DHIS. The overall conclusion of the investigation is that given their context specific issues, when LMICs are involved in the GSD work practices that are taken-for-granted or left on background in developed countries need to be highlighted or visualized. Issues such as lack of skills, cultural diversity and financial resources to hire skilled people or provide access to tools developed to support cooperation, affects more LMICs than developed countries.

Through this research we were able to make visible the task-artefact fit work practices that are usually left in the background. Thus, software customization has much to contribute to GSD since it can provide an avenue to explore more deeply the varied and invisible work practices. This potential will only be realized as GSD field is distributed or include alliances from communities with cultural, social, technological and economic diversification, as was the case of this research.

REFERENCES

1. Sahay, S., Global software alliances: the challenge of 'standardization'. *Scandinavian Journal of Information Systems*, 2003. 15(1): p. 3-21.
2. Conchuir, E.O., et al. Exploring the assumed benefits of global software development. in *International Conference on Global Software Engineering, ICGSE'06*. 2006. IEEE.
3. Layman, L., et al., Essential communication practices for Extreme Programming in a global software development team. *Information and software technology*, 2006. 48(9): p. 781-794.
4. Nardi, B.A., *A Small Matter of Programming*. 1993, Massachusetts: The MIT Press.
5. Star, S.L. and A. Strauss, *Layers of silence, arenas of voice: The ecology of visible and invisible work*. *Computer supported cooperative work (CSCW)*, 1999. 8(1-2): p. 9-30.
6. Oudshoorn, N., *Diagnosis at a distance: the invisible work of patients and healthcare professionals in cardiac telemonitoring technology*. *Sociology of health & illness*, 2008. 30(2): p. 272-288.
7. Schmidt, K. and L. Bannon, *Taking CSCW seriously*. *Computer Supported Cooperative Work (CSCW)*, 1992. 1(1-2): p. 7-40.
8. Suchman, L., *Supporting Articulation Work*, in *Computerization and Controversy: Value Conflicts and Social Choices*, R. Kling, Editor. 1996, Academic Press: San Diego, USA. p. 407-423.
9. Schmidt, K. and C. Simonee, *Coordination mechanisms: Towards a conceptual foundation of CSCW systems design*. *Computer Supported Cooperative Work (CSCW)*, 1996. 5(2-3): p. 155-200.
10. Daniels, H., et al., *Activity Theory in Practice: Promoting Learning Across Boundaries and Agencies*. 2010, London and New York: Routledge Taylor & Francis Group.
11. Suchman, L.A., *Plans and situated actions: the problem of human-machine communication*. 2007: Cambridge University Press.
12. Orlikowski, W.J., *Practice in research: phenomenon, perspective and philosophy*, in *Cambridge Handbook of Strategy as Practice*, D. Golsorkhi, et al., Editors. 2010, Cambridge University Press: Cambridge, England. p. 23-33.
13. Clement, A., *Looking for the designers: Transforming the 'invisible' infrastructure of computerised office work*. *AI & society*, 1993. 7(4): p. 323-344.