Some Non-Usage Data for a Distributed Editor: The Saros Outreach

Lutz Prechelt
Freie Universität Berlin, Institut f. Informatik, Berlin, Germany
prechelt@inf.fu-berlin.de

ABSTRACT

After contacting more than 40 companies and 11 OSS projects regarding using our distributed editor Saros, we find that almost all of those many who have a use case for it, are reluctant to even try it out. It appears that distance matters even by anticipatory obedience.

Categories and Subject Descriptors

K.7.m [**The Computing Profession**]: Miscellaneous

General Terms

Human Factors

Keywords

Outreach, Eclipse, Collaboration, Distributed Development

1. INTRODUCTION

Saros (www.saros-project.org) is an Open Source Eclipse plugin for distributed collaborative editing and viewing. This means that two or more participants of a Saros session have an identical copy of all files of a project (using any textual languages) and any change made by any participant of the session will be reproduced in real-time in the corresponding files (and, if applicable, on the screen) of all other participants. Saros shows where each participant is working and what were the last few changes made by each. One can set Saros to follow the view seen by another participant automatically in order to watch what s/he is doing or showing. All navigation and editing functions of Eclipse (including refactorings) are still available. A VoIP connection completes the collaboration scenario.

Saros is thus applicable to various work modes, including joint viewing of code, distributed pair programming, and distributed side-by-side programming and is therefore potentially useful for any organization performing distributed software development that uses Eclipse.

2. SAROS OUTREACH RESULTS

We have been developing Saros since 2006 and it is by now approaching full industry-readiness. Since July 2010 we have begun to look for partners who want to use Saros and will let us perform field research on its process implications.

CHASE'11, May 21, 2011, Waikiki, Honolulu, HI, USA
ACM 978-1-4503-0576-1/11/05

Open Source projects: We started by contacting Open Source (OSS) projects, because we expected them to have the highest need (due to the super-distributed fashion in which they work) and to provide the most unbureaucratic access. We spent a total of about 3 person months of effort in contacts with 11 selected Open Source projects, with these main results: **(1)** We often did not even manage to produce any answer to our contact requests at all. **(2)** Lack of sufficient Eclipse usage is usually a showstopper in non-Java projects and often even in Java projects. Switching to Eclipse is almost never considered. **(3)** The language barrier between developers also appears to be a substantial issue. **(4)** The work days and times of suitable pairs of developers did often not overlap. **(5)** A good fraction of respondents refused pair programming and identified Saros with nothing else.

The net effect was apparently zero; not a single project took up regular use of Saros.

Multipliers: We then contacted multipliers and companies, usually by personalized email, sometimes based on previous acquaintance. Of the 12 multipliers, 6 were SE researchers and had essentially no effect. The other 6 were self-employed consultants from the agile methods community: 3 never replied; 1 is not interested in tools generally; 1 works in a non-Eclipse context. The only success was Kent Beck, who says "This is clearly the future of pair programming."

Companies: We contacted more than 40 companies, large and small, that we found likely to have use cases for Saros. Of these contacts, 11 are in too early a stage to tell (4 look promising); 8 never gave any reply whatsoever ("black hole"); 4 answered once or twice and then turned into black holes; 11 saw no use case (e.g. lack of remote collaboration, lack of Eclipse use), only 2 of these sounded really interested; 5 had different priorities (no capacity to evaluate Saros); 1 is interested but misses one essential feature; 1 is interested but has network policy issues; only 2 have actually taken up Saros use (on a small scale).

3. CONCLUSION

In our attempts to find partners for field research regarding Saros, we observed that it is *much* more difficult than expected to find people who will actually use such functionality. Our data suggest the reasons are strongly human, not technical. Olson and Olson are right: "Distance matters". As much as co-located software developers love to collaborate informally, synchronous collaboration in a distributed setting appears to be a very different issue.