Onboarding in Open Source Software Projects: A Preliminary Analysis

Jürgen Münch*
with Fabian Fagerholm*, Patrik Johnson*,
Alejandro Sanchez Guinea*, Jay Borenstein†

*University of Helsinki
†Stanford University
Facebook

26 August 2013
VirtuES Workshop (colocated with ICSE’11)

Overview
- Motivation
- Context
- Research Approach
- Results
- Limitations
- Conclusions and Future work
Motivation

- **Most companies need to develop their software with so-called software ecosystems**
  - reasons: software size exceeds capabilities of an individual organization, need to combine own offerings with infrastructure(s) of ecosystem, market dominance of specific ecosystem...

- **OSS projects (such as Android, LAMP stack) play a major role as software ecosystems**
  - similar to Global Software Development (GSD) in many ways
  - projects are often highly decentralised
  - with participants from a wide range of geographical locations and cultural backgrounds

- **But the development process followed in many OSS projects is specific**
  - Developers have a large degree of freedom to decide how to manage themselves within their projects

Motivation

- **Companies need to get involved in OSS projects quickly**
  - how to support the entry of new members into OSS projects?
  - guidance to address this is widely missing

- **OSS development teams are typically self-organized virtual teams**
  - lack of formally appointed leaders or indications of rank or role
  - shared power
  - self-assignment
  - "soft delegation"

- **Onboarding or Organizational Socialization**
  - "the process that helps new employees learn the knowledge, skills, and behaviours they need to succeed in their new organizations" [1]
  - potential impact factors on the effectiveness and efficiency of onboarding: mentoring, project type, face-to-face meetings, ...
  - Question: What is the impact of mentoring in OSS projects?
Context

- Facebook's Education Modernization program 2013
- 9 open source projects
- 130 students, > 12 Universities
- Developers submitted their OSS project preferences
- Based in this they were assigned to virtual teams with 4–8 participants from at least two different universities
- Each OSS project had two separate virtual teams working for it, and provided an experienced senior developer to mentor the developer teams
- Onboarding support: kickoff and continuous mentoring in the project
- Focus of this study: 4 OSS projects with participation of University of Helsinki through the Software Factory Laboratory
Context

TABLE I. PARTICIPATING OPEN SOURCE PROJECTS.

<table>
<thead>
<tr>
<th>Project name</th>
<th>URL</th>
<th>Included in this study?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freeseer</td>
<td><a href="http://freeseer.github.io/">http://freeseer.github.io/</a></td>
<td>No</td>
</tr>
<tr>
<td>Kotlin</td>
<td><a href="http://kotlin.jetbrains.org/">http://kotlin.jetbrains.org/</a></td>
<td>Yes</td>
</tr>
<tr>
<td>MongoDB</td>
<td><a href="http://www.mongodb.org/">http://www.mongodb.org/</a></td>
<td>No</td>
</tr>
<tr>
<td>Mozilla OpenBadges</td>
<td><a href="http://openbadges.org/">http://openbadges.org/</a></td>
<td>No</td>
</tr>
<tr>
<td>ReviewBoard</td>
<td><a href="http://www.reviewboard.org/">http://www.reviewboard.org/</a></td>
<td>No</td>
</tr>
<tr>
<td>Phabricator</td>
<td><a href="http://phabricator.org/">http://phabricator.org/</a></td>
<td>Yes</td>
</tr>
<tr>
<td>PouchDB</td>
<td><a href="http://pouchdb.com/">http://pouchdb.com/</a></td>
<td>No</td>
</tr>
<tr>
<td>Ruby on Rails</td>
<td><a href="http://rubyonrails.org/">http://rubyonrails.org/</a></td>
<td>Yes</td>
</tr>
<tr>
<td>Socket.IO</td>
<td><a href="http://socket.io/">http://socket.io/</a></td>
<td>Yes</td>
</tr>
</tbody>
</table>

Kickoff

- Three-day kickoff event at Facebook’s Menlo Park headquarters
- Students got acquainted with other members of their virtual teams and their open source mentors
- Mentors provided practical training
Research Approach

- **Goal:**
  - characterise and understand the onboarding process in the OSS projects under study, with particular consideration of the role of mentoring as part of that process
  - ultimate aim is to derive guidelines for onboarding, e.g., how to design a mentoring program

- **Focus:**
  - contribution level over time, i.e., productivity and communication activity among team members

- **Design:**
  - quantitative multiple-case study approach
  - compare a subset of the developers receiving mentoring support against developers in the same OSS projects who did not receive the mentoring and enter OSS projects in the “usual way”

- **Sample:**
  - random sample of 20 developers for each group
  - first 12 weeks of the project

---

GQM Plan

**TABLE II. DEFINITION OF GQM MEASUREMENT GOAL.**

<table>
<thead>
<tr>
<th>Object</th>
<th>Onboarding process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Characterisation and Understanding</td>
</tr>
<tr>
<td>Focus</td>
<td>Contribution level over time</td>
</tr>
<tr>
<td>Viewpoint / Stakeholder</td>
<td>Project manager / Open source mentor</td>
</tr>
<tr>
<td>Context</td>
<td>International collaboration project (see main text)</td>
</tr>
</tbody>
</table>

**Q1** How much time does a developer communicate in the project over time?

**Q2** How much time and effort does a developer put into the project over time?

**Q3** How much does a developer contribute to the project over time?

**Q4** How much mentoring does each mentor give to their respective team(s)?

**Q5** What aspects of onboarding are considered important by project mentors and developers?

**Q6** What is the influence of mentoring on the performance of the developers?
Measures

- **Aggregate Measure „Activity“**
  - Activity = #commits + #pull_requests + #interactions
    (inspired by Fritz et al. [2])

- **Commit**
  - submission of the latest changes in the source code to the repository in the context of revision control systems

- **Pull Requests**
  - a notification that is made to others about changes pushed to the repository, so that interested parties can review changes, discuss potential modifications and push follow-up commits if necessary (as characterized by GitHub) [3]

- **Interaction**
  - a single message posted in a GitHub discussion forum

Preliminary Results

**TABLE III. TOTAL ACCUMULATED ACTIVITY.**

<table>
<thead>
<tr>
<th>Developers</th>
<th>Commits</th>
<th>Pull requests</th>
<th>Interactions</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mentored</td>
<td>64</td>
<td>111</td>
<td>251</td>
<td>426</td>
</tr>
<tr>
<td>Non-mentored</td>
<td>19</td>
<td>37</td>
<td>92</td>
<td>148</td>
</tr>
</tbody>
</table>

- Developers in the mentored group show roughly three times more activity than the non-mentored group
Preliminary Results

![Graph showing activity over weeks with annotations for same level, three-week plateau, significant increase, and steady increase.]

GQM Plan

**TABLE II. DEFINITION OF GQM MEASUREMENT GOAL.**

<table>
<thead>
<tr>
<th>Object</th>
<th>Onboarding process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Characterisation and Understanding</td>
</tr>
<tr>
<td>Focus</td>
<td>Contribution level over time</td>
</tr>
<tr>
<td>Viewpoint / Stakeholder</td>
<td>Project manager / Open source mentor</td>
</tr>
<tr>
<td>Context</td>
<td>International collaboration project (see main text)</td>
</tr>
</tbody>
</table>

**Q1** How much time does a developer communicate in the project over time?

**Q2** How much time and effort does a developer put into the project over time?

**Q3** How much does a developer contribute to the project over time?

**Q4** How much mentoring does each mentor give to their respective team(s)?

**Q5** What aspects of onboarding are considered important by project mentors and developers?

**Q6** What is the influence of mentoring on the performance of the developers?
Results form the Ruby on Rails Project

- **Speeding up Ruby on Rails**
  - Students have been working on the Active Record database access component.
  - Active Record is an object-relational mapper that allows for abstracting an underlying relational database. Thus developers do not need to write database queries directly.
  - The students implemented a statement cache for Active Record. The statement cache makes Active Record faster by reusing data structures that would otherwise be re-generated on each database query.
  - The new statement cache has been merged into the release of Ruby on Rails 4.0, which is a major update to the web framework.

Limitations of this Preliminary Study

- We did not distinguish between mentoring and other onboarding support factors (such as kickoff event)
- We included only part of the data available (4 of 9 projects, only first 12 weeks)
- We used student subjects
- We have not examined whether developers in the non-supported group have received any treatment during the observed period of time that would impact the results
- We have not statistically evaluated the construct validity of our activity metric
- The subjects may have altered their behaviour in response to the study
Summary and Future Research

- We thus find support that onboarding support increases the chance of developers to be exposed to, select, and perform tasks in a proactive and self-directed manner in open source projects.
- We assume that this applies in other kinds of GSD settings where the organisational and team structure is similar.

Future research
- Expansion of the sample
- Analyzing commits, pull requests, and interactions over time
- Developing more comprehensive measures
- Developing a more comprehensive set of measures
- Taking other impact and context factors into account (such as project type, mentoring type, etc.)

We assume that mentoring is highly important for maintaining developers' motivation and for attaining good cohesion within virtual teams.

Contact and Further Information

@Juergen_Muench

- Software Systems Engineering Research Group
  [http://www.sserg.org](http://www.sserg.org)
- Software Factory
  [http://www.softwarefactory.cc](http://www.softwarefactory.cc)
- Software Factory video
  [http://www.youtube.com/watch?v=uPAE1uRP65Y](http://www.youtube.com/watch?v=uPAE1uRP65Y)
- Software Factory Speeds up Ruby on Rails (article)
  [http://www.sserg.org/software-factory-speeds-up-ruby-on-rails/](http://www.sserg.org/software-factory-speeds-up-ruby-on-rails/)
Reference of the Article


The article can be found online here:

Further References

