Barriers faced by newcomers to Open Source Software projects

Igor Steinmacher

Marco Aurélio Gerosa

ICMC USP SÃO CARLOS

IME-USP

EACH USP

DC Departamento de Computação

INSTITUTO DE INFORMÁTICA UFG

NAPSOL NÚCLEO DE APOIO À PESQUISA EM SOFTWARE LIVRE

Context

“Newcomers are explorers who must orient themselves within an unfamiliar landscape...”

[Degenais et al. 2010]

... and need support on their first steps

B. Dagenais, H. Ossher, R.K.E Bellamy, M.P. Robillard and J.P. de Vries, **Moving into a new software project landscape**, in ICSE 2010.

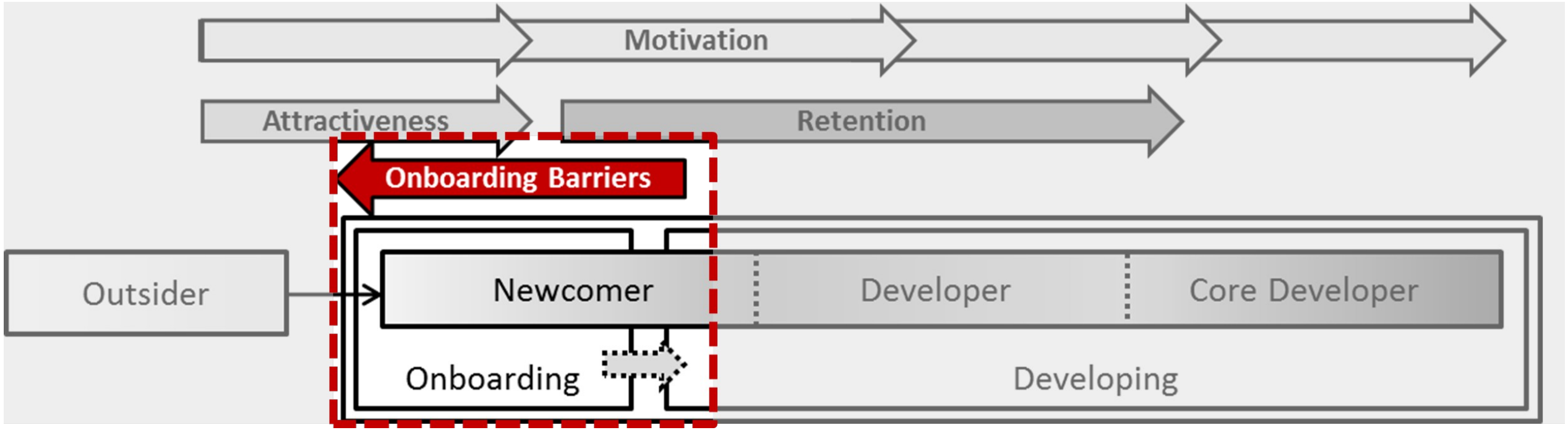
Context

- Many Open Source Software (OSS) projects rely on contributions from volunteers
- It is essential to motivate, engage, and retain new participants in order to promote a sustainable community
- However...

Context

- Newcomers face barriers to place their first contribution in OSS projects
  - Newcomers are expected to learn about the projects their own
  - We are talking about volunteers
  - Literature focuses on long term commitment/retention
  - What about those short-term (single time) contributors?

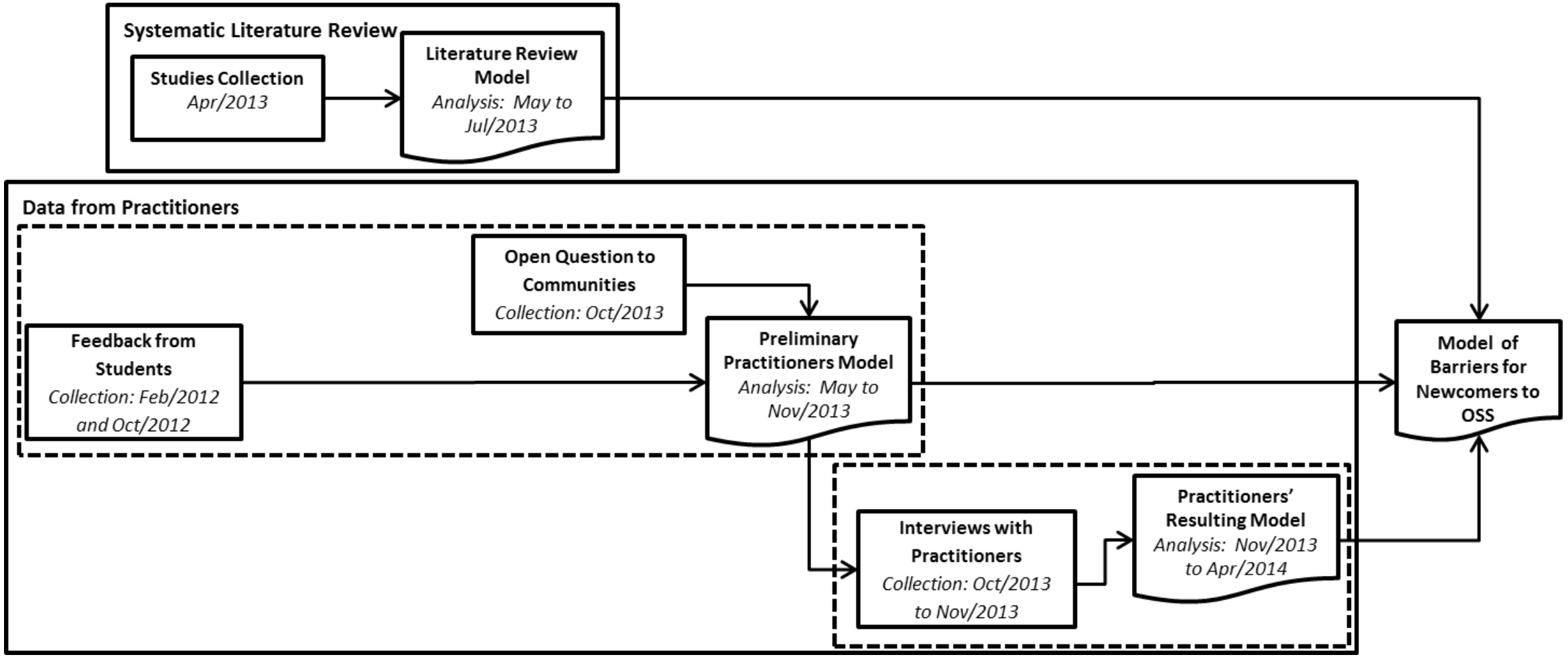
Context


Motivation
Attractiveness
Retention
Onboarding Barriers
Outsider
Newcomer
Developer
Core Developer
Onboarding
Developing

Goal

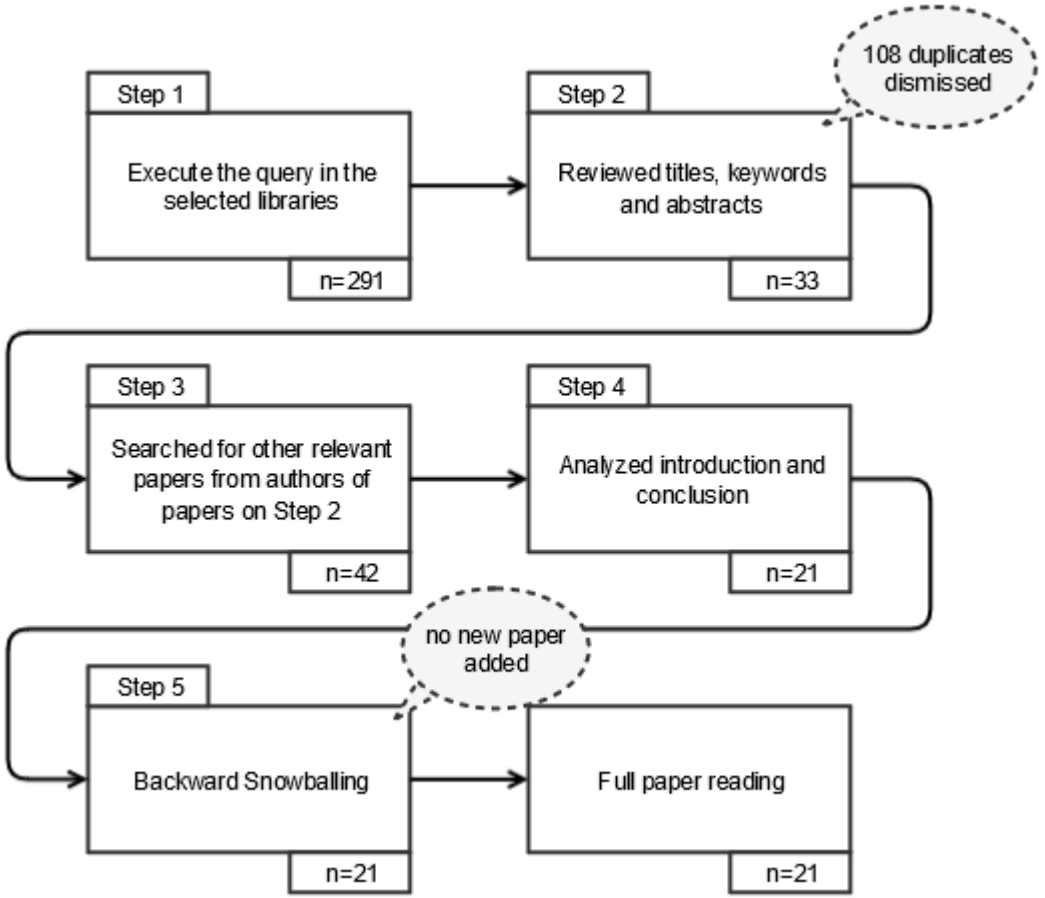
Empirically **identify** and **categorize** the **barriers** faced by **newcomers** when **placing their first contribution** to **OSS projects**

Method


Systematic Literature Review
Studies Collection
Apr/2013
Literature Review Model
Analysis: May to Jul/2013
Data from Practitioners
Open Question to Communities
Collection: Oct/2013
Feedback from Students
Collection: Feb/2012 and Oct/2012
Preliminary Practitioners Model
Analysis: May to Nov/2013
Interviews with Practitioners
Collection: Oct/2013 to Nov/2013
Practitioners' Resulting Model
Analysis: Nov/2013 to Apr/2014
Model of Barriers for Newcomers to OSS

Method - SLR

- RQ: *What are the barriers that influence newcomers' onboarding to OSS projects?*
  - Search on digital libraries: ACM, IEEE, Scopus, and Springer Link
  - Author snowballing + Backward snowballing (single step)



Method – data from practitioners (collection)

- Three sources:
  - **Source 1:** feedback from students that contributed to OSS projects
  - **Source 2:** answers to an open question sent to developers' mailing lists of OSS projects
  - **Source 3:** semi-structured interviews conducted with newcomers and members of OSS projects.

Method – data from practitioners (collection)

- Source 1: feedback from students
  - Assignment: contribute to an OSS projects
  - Feedback: open-ended questionnaire
  - 9 subjects (5 PhD candidates / 4 undergrad students)
  - 3 different projects

Method – data from practitioners (collection)

- Source 2: answers to a questionnaire sent to OSS developers
  - Recruitment: mailing lists and forums
  - Question: *“In your opinion, what are the main difficulties faced by newcomers when they want to start contributing to this project? (Consider technical and non-technical issues).”*
  - 24 complete answers / 9 different projects

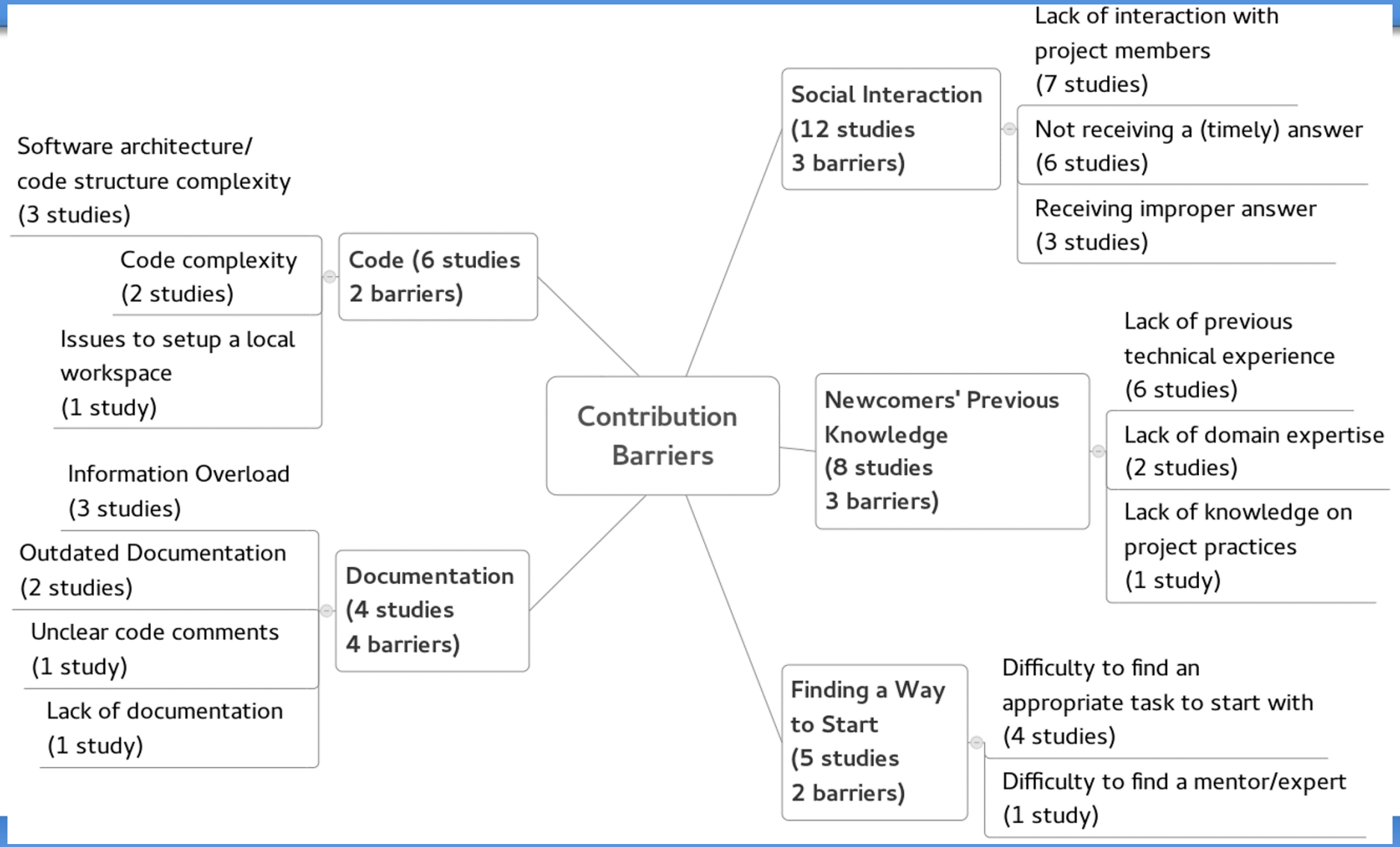
Method – data from practitioners (collection)

- Source 3: semi-structured interviews with practitioners
  - complement the findings gathered from Sources 1 and 2
  - subjects that belong to four different profiles:
    - **Experienced members**
    - **Newcomers that succeeded**
    - **Dropout Newcomers**
    - **Onboarding Newcomers**
  - Recruited via mailing lists and direct contact (depending on the profile)

Method – data from practitioners (analysis)

- Procedures of Grounded Theory
  - Open coding
  - Axial coding
- Analysis occurred in 2 steps
  - Step 1: Sources 1 and 2 were analyzed
  - Step 2: categories from Step 1 used as seeds to analysis of Source 3

Results – SLR


Contribution Barriers
Code (6 studies 2 barriers)
Code complexity (2 studies)
Software architecture/ code structure complexity (3 studies)
Issues to setup a local workspace (1 study)
Documentation (4 studies 4 barriers)
Information Overload (3 studies)
Outdated Documentation (2 studies)
Unclear code comments (1 study)
Lack of documentation (1 study)
Social Interaction (12 studies 3 barriers)
Lack of interaction with project members (7 studies)
Not receiving a (timely) answer (6 studies)
Receiving improper answer (3 studies)
Newcomers' Previous Knowledge (8 studies 3 barriers)
Lack of previous technical experience (6 studies)
Lack of domain expertise (2 studies)
Lack of knowledge on project practices (1 study)
Finding a Way to Start (5 studies 2 barriers)
Difficulty to find an appropriate task to start with (4 studies)
Difficulty to find a mentor/expert (1 study)

Results – Students + survey

Category	# of barriers
Social interaction issues	4
Newcomers' behavior	2
Newcomers' technical knowledge	5
Finding a way to start	3
Documentation problems	10
Code issues	5
Issues setting up workspace	4

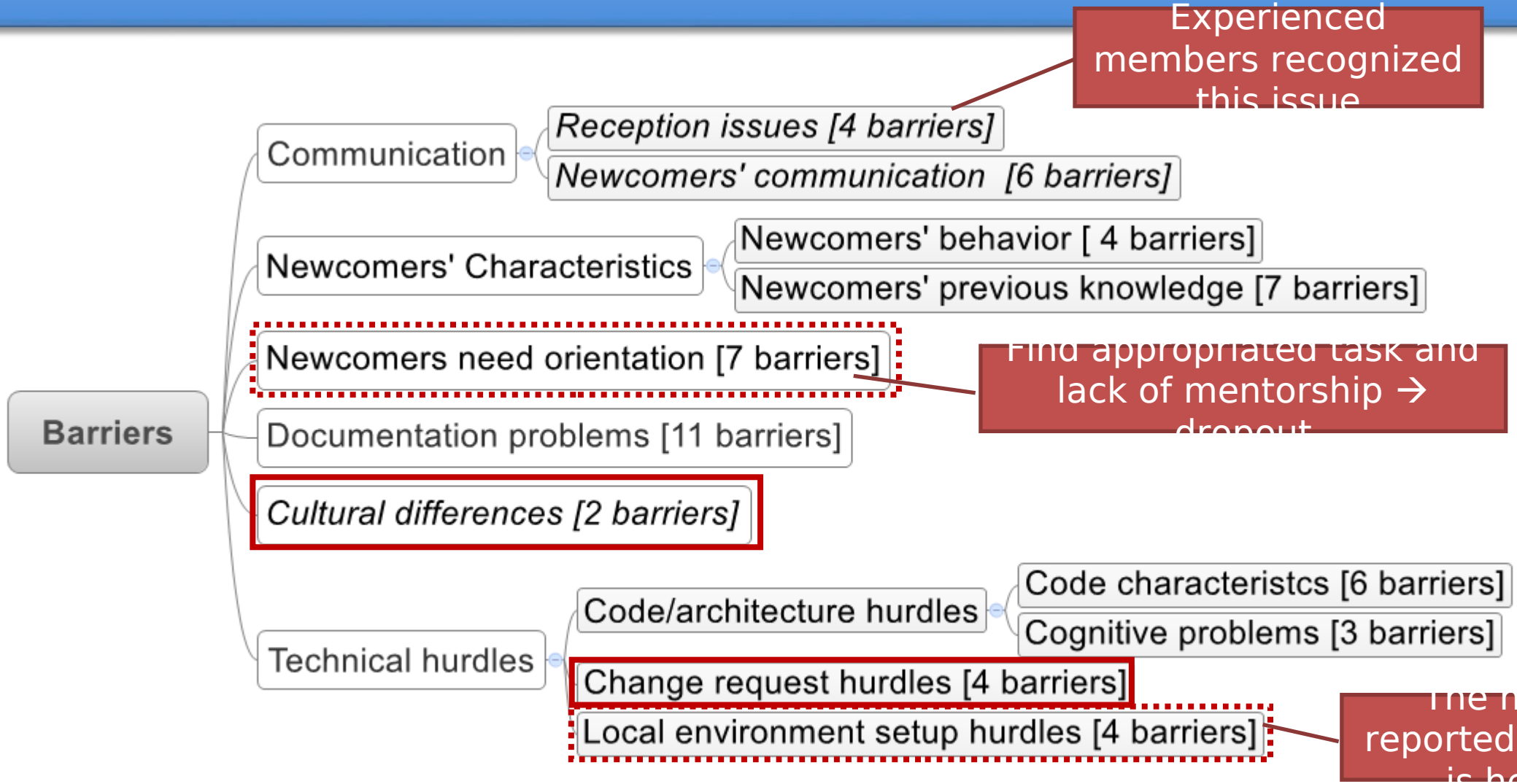
Reception issues reported only by newcomers

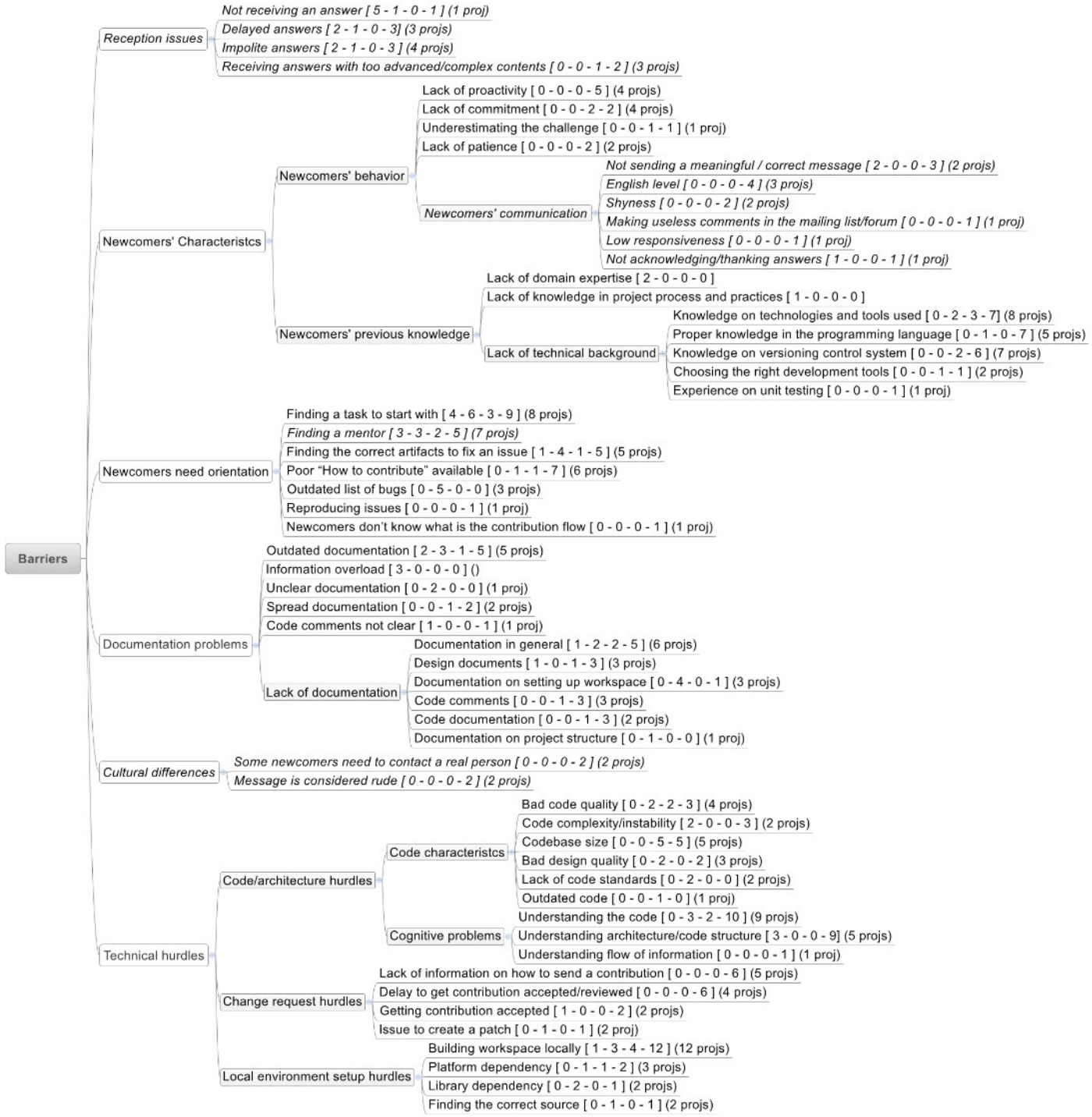
In contrast, reported mainly by 'old' members

Lack of/outdated documentation

Complaints from students

Results – merging interviews




Barriers
Reception issues
Not receiving an answer [5 - 1 - 0 - 1] (1 proj)
Delayed answers [2 - 1 - 0 - 3] (3 projs)
Impolite answers [2 - 1 - 0 - 3] (4 projs)
Receiving answers with too advanced/complex contents [0 - 0 - 1 - 2] (3 projs)
Newcomers' Characteristcs
Newcomers' behavior
Lack of proactivity [0 - 0 - 0 - 5] (4 projs)
Lack of commitment [0 - 0 - 2 - 2] (4 projs)
Underestimating the challenge [0 - 0 - 1 - 1] (1 proj)
Lack of patience [0 - 0 - 0 - 2] (2 projs)
Newcomers' communication
Not sending a meaningful / correct message [2 - 0 - 0 - 3] (2 projs)
English level [0 - 0 - 0 - 4] (3 projs)
Shyness [0 - 0 - 0 - 2] (2 projs)
Making useless comments in the mailing list/forum [0 - 0 - 0 - 1] (1 proj)
Low responsiveness [0 - 0 - 0 - 1] (1 proj)
Not acknowledging/thanking answers [1 - 0 - 0 - 1] (1 proj)
Newcomers' previous knowledge
Lack of domain expertise [2 - 0 - 0 - 0]
Lack of knowledge in project process and practices [1 - 0 - 0 - 0]
Lack of technical background
Knowledge on technologies and tools used [0 - 2 - 3 - 7] (8 projs)
Proper knowledge in the programming language [0 - 1 - 0 - 7] (5 projs)
Knowledge on versioning control system [0 - 0 - 2 - 6] (7 projs)
Choosing the right development tools [0 - 0 - 1 - 1] (2 projs)
Experience on unit testing [0 - 0 - 0 - 1] (1 proj)
Newcomers need orientation
Finding a task to start with [4 - 6 - 3 - 9] (8 projs)
Finding a mentor [3 - 3 - 2 - 5] (7 projs)
Finding the correct artifacts to fix an issue [1 - 4 - 1 - 5] (5 projs)
Poor "How to contribute" available [0 - 1 - 1 - 7] (6 projs)
Outdated list of bugs [0 - 5 - 0 - 0] (3 projs)
Reproducing issues [0 - 0 - 0 - 1] (1 proj)
Newcomers don't know what is the contribution flow [0 - 0 - 0 - 1] (1 proj)
Documentation problems
Outdated documentation [2 - 3 - 1 - 5] (5 projs)
Information overload [3 - 0 - 0 - 0] ()
Unclear documentation [0 - 2 - 0 - 0] (1 proj)
Spread documentation [0 - 0 - 1 - 2] (2 projs)
Code comments not clear [1 - 0 - 0 - 1] (1 proj)
Lack of documentation
Documentation in general [1 - 2 - 2 - 5] (6 projs)
Design documents [1 - 0 - 1 - 3] (3 projs)
Documentation on setting up workspace [0 - 4 - 0 - 1] (3 projs)
Code comments [0 - 0 - 1 - 3] (3 projs)
Code documentation [0 - 0 - 1 - 3] (2 projs)
Documentation on project structure [0 - 1 - 0 - 0] (1 proj)
Cultural differences
Some newcomers need to contact a real person [0 - 0 - 0 - 2] (2 projs)
Message is considered rude [0 - 0 - 0 - 2] (2 projs)
Technical hurdles
Code/architecture hurdles
Code characteristcs
Bad code quality [0 - 2 - 2 - 3] (4 projs)
Code complexity/instability [2 - 0 - 0 - 3] (2 projs)
Codebase size [0 - 0 - 5 - 5] (5 projs)
Bad design quality [0 - 2 - 0 - 2] (3 projs)
Lack of code standards [0 - 2 - 0 - 0] (2 projs)
Outdated code [0 - 0 - 1 - 0] (1 proj)
Cognitive problems
Understanding the code [0 - 3 - 2 - 10] (9 projs)
Understanding architecture/code structure [3 - 0 - 0 - 9] (5 projs)
Understanding flow of information [0 - 0 - 0 - 1] (1 proj)
Change request hurdles
Lack of information on how to send a contribution [0 - 0 - 0 - 6] (5 projs)
Delay to get contribution accepted/reviewed [0 - 0 - 0 - 6] (4 projs)
Getting contribution accepted [1 - 0 - 0 - 2] (2 projs)
Issue to create a patch [0 - 1 - 0 - 1] (2 proj)
Local environment setup hurdles
Building workspace locally [1 - 3 - 4 - 12] (12 projs)
Platform dependency [0 - 1 - 1 - 2] (3 projs)
Library dependency [0 - 2 - 0 - 1] (2 projs)
Finding the correct source [0 - 1 - 0 - 1] (2 projs)

Discussion

- Literature mostly reports quantitative evidences based on historical data
  - Focus on socialization
- Newcomers mainly complain of reception, issues to setup the workspace locally and code understanding
  - Technical problems → Students and onboarding newcomers
- Experienced members
  - Recognize some barriers
  - Some pointed newcomers as the only problem (knowledge, commitment)

Discussion

- 50 out of 58 barriers presented in the resulting model were identified in the data from interviews with practitioners

- Less than 30% of the barriers (17 barriers) were evidenced by the literature

Conclusions

- Placing a first contribution in an OSS project can be a tough task
- There are technical, social and process barriers that newcomers need to overcome
- These results provides insights for communities that want to smooth newcomer onboarding
- Lays a foundation for building better onboarding support tools

Conclusions

- First step towards better orienting newcomers' first steps
- A smooth first contribution may increase the total number of successful contributions
- Maps or signs to orient newcomers and guide them, or, at least, warn them about the barriers they can find
- We believe that simple actions can make a great impact

On going research

- Portal organizing information according to the barriers model was conceived (flosscoach.com)
  - Students received an assignment to contribute to specific projects
  - Use of diaries → qualitative analysis
  - Case x control
  - How can the portal support newcomers?

Thanks!

QUESTIONS?