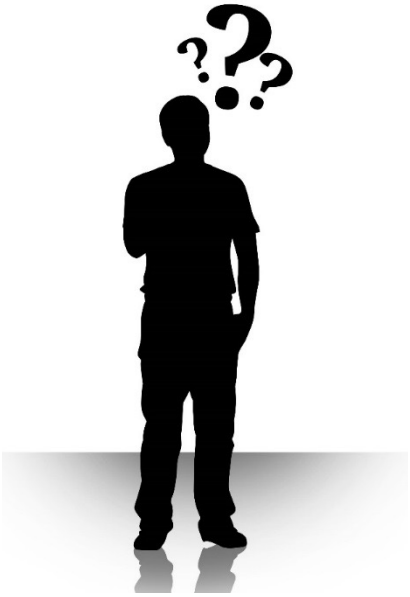
Supporting Newcomers in Software Development Projects



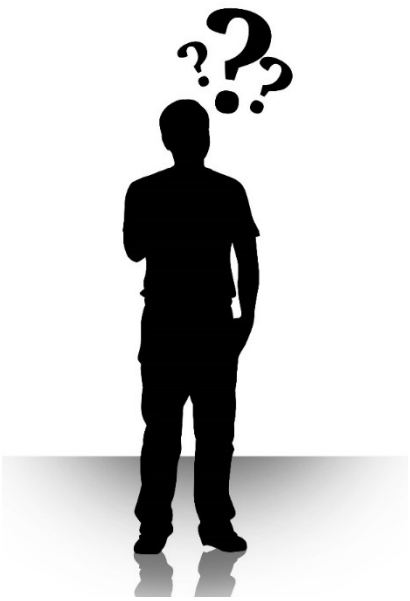
Doctoral Dissertation
by
Sebastiano Panichella

Under the supervision of:

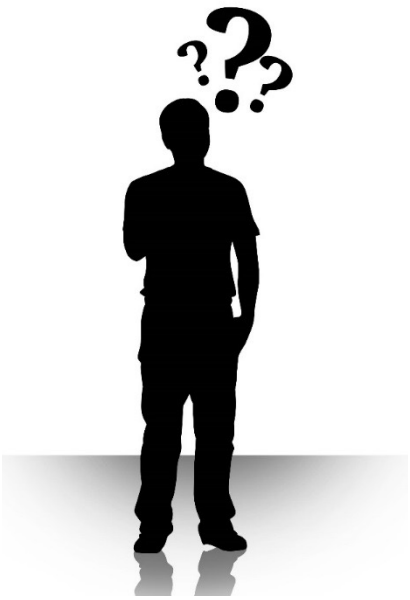
Prof. Massimiliano Di Penta
Prof. Gerardo Canfora

July 2014

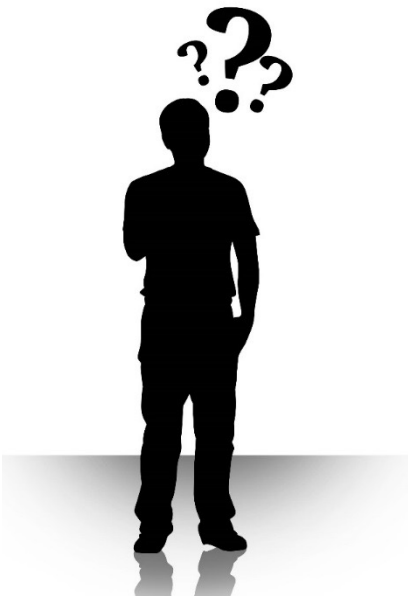
Newcomer Learning Path…



Newcomer Learning Path…



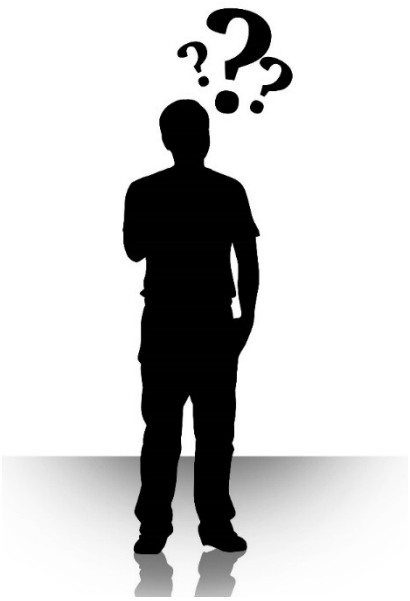
Newcomer Learning Path…



Data Extraction

- Versioning Systems
- Mailing Lists
- Issue trackers
- Q&A site (e.g. StackOverflow)

Newcomer Learning Path…

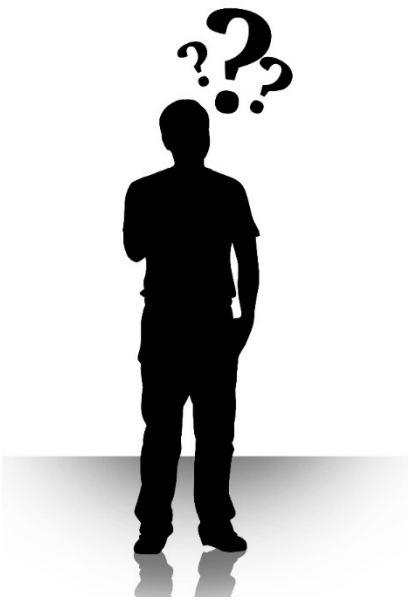


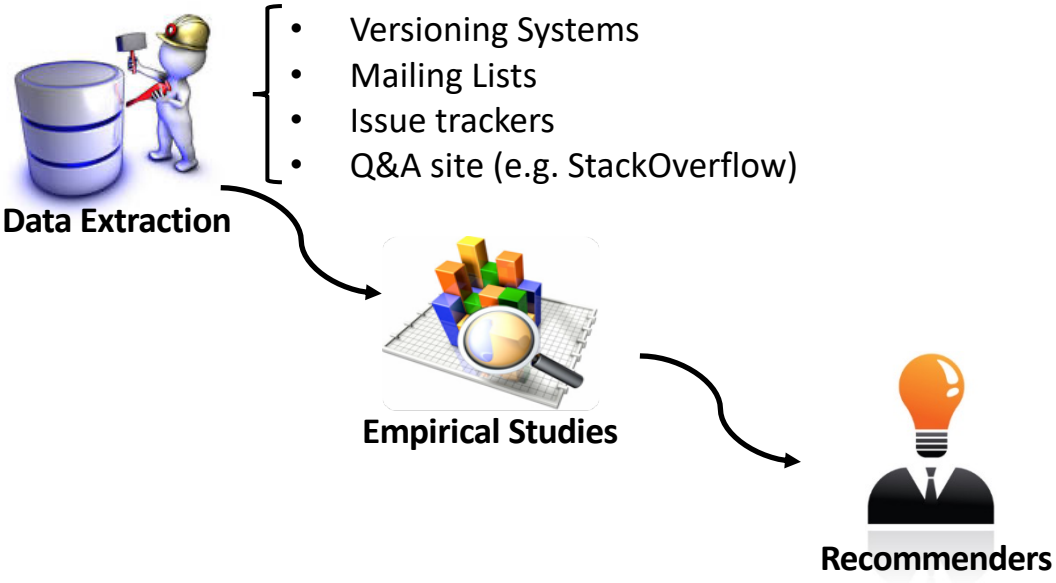
Data Extraction

- Versioning Systems
- Mailing Lists
- Issue trackers
- Q&A site (e.g. StackOverflow)

Empirical Studies

Newcomer Learning Path…





Newcomer Training Process:

Studies

Data Extraction

Recommenders

Newcomer Training Process:

Studies

Mentoring

Data Extraction

Recommenders

1) Recommend Mentors

Newcomer Training Process:

2) Analyze Software Artifacts:
c) investigate how newcomers browse artifacts software
d) investigate how newcomers generate source code summaries

Studies

Mentoring ⟷ **Perform Development Tasks**

Data Extraction

Recommenders

1) Recommend Mentors

2) Supporting Source Code Comprehension and Re-documentation

Newcomer Training Process:

Studies

2) Analyze Software Artifacts:
c) investigate how newcomers browse artifacts software
d) investigate how newcomers generate source code summaries
3) Analyze developers:
c) social activity
d) technical activity

Mentoring ↔ **Perform Development Tasks** ↔ **Team Collaborations**

Data Extraction

Recommenders

1) Recommend Mentors
2) Supporting Source Code Comprehension and Re-documentation
3) Recommend Refactoring

Newcomer Training Process:

Studies

2) Analyze Software Artifacts:
c) generate source code summaries
d) support maintenance tasks

3) Analyze developers:
c) social activity
d) technical activity

Mentoring ↔ **Perform Development Tasks** ↔ **Team Collaborations**

Data Extraction

Recommenders

1) Recommend Mentors

2) Supporting Source Code Comprehension and Re-documentation

3) Recommend Refactoring

Thesis Structure

- PART I

- PART II

- PART III

Thesis Structure

- PART I: analyzing data from software repositories to support team work.

- PART II

- PART III

Thesis Structure

- PART I: analyzing data from software repositories to support team work.

- PART II: analyzing how developers use software artifacts to help newcomers in program comprehension task.

- PART III

Thesis Structure

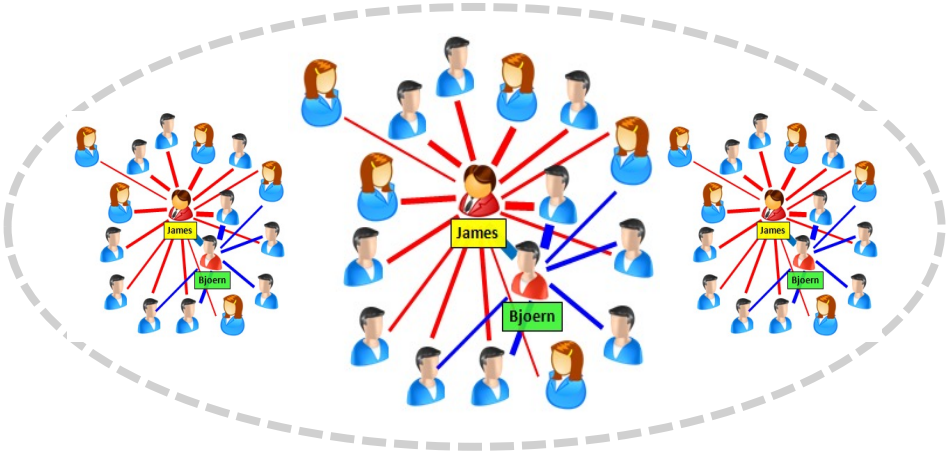
- PART I: analyzing data from software repositories to support team work.

- PART II: analyzing how developers use software artifacts to help newcomers in program comprehension task.

- PART III: developing recommenders to support concretely project newcomers.

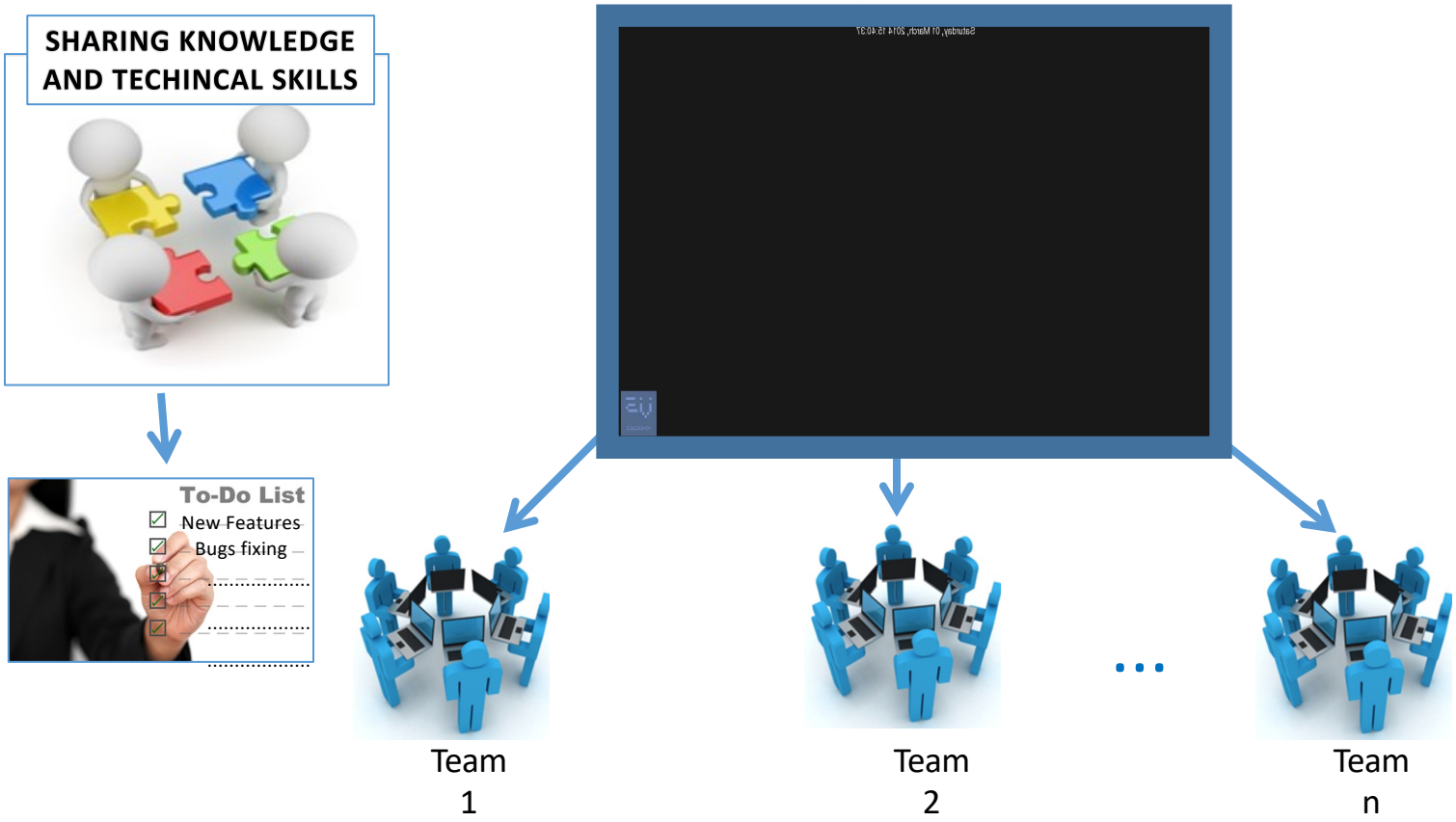
PART I

Analysis of Developers' Communication



Emerging Teams in Open Source Projects

https://code.google.com/p/gource/



Socio-Technical Congruence in Developers Social Networks

Latent Social Structure in Open Source Projects

Christian Bird, David Pattison, Raissa D'Souza,
Vladimir Filkov and Premkumar Devanbu
Dept. of Computer Science, Kemper Hall,
University of California, Davis, CA, USA,
cabird,dspattison,rmdsouza,vfilkov,ptdevanbu@ucdavis.edu

ABSTRACT

Commercial software project managers design project organizational structure carefully, mindful of available skills, division of labour, geographical boundaries, etc. These organizational "cathedrals" are to be contrasted with the "bazaar-like" nature of Open Source Software (OSS) Projects, which have no pre-designed organizational structure. Any structure that exists is dynamic, self-organizing, latent, and usually not explicitly stated. Still, in large, complex, successful, OSS projects, we do expect that subcommunities will form spontaneously within the developer teams. Studying these subcommunities, and their behavior can shed light on how successful OSS projects self-organize. This phenomenon could well hold important lessons for how commercial software teams might be organized. Building on known well-established techniques for detecting *community structure* in complex networks, we extract and study latent subcommunities from the email social network of several projects: Apache HTTPD, Python, PostgresSQL, Perl, and Apache ANT. We then validate them with software development activity history. Our results show that subcommunities do indeed spontaneously arise within these projects as the projects evolve. These subcommunities manifest most strongly in technical discussions, and are significantly connected with collaboration behaviour.

Categories and Subject Descriptors

D.2.9 [Software Engineering]: Management—*programming teams*; D.2.8 [Software Engineering]: Metrics—*process metrics*

General Terms

Human Factors, Measurement, Management

Keywords

Open Source Software, social networks, collaboration

This work was supported by a grant from the National Science Foundation Grant no. NSF-SoD-0613049 and software donations from SciTools and GraphmaTech Corporations.

SIGSOFT 2008/FSE 16, November 9–15, Atlanta, Georgia, USA

1. INTRODUCTION

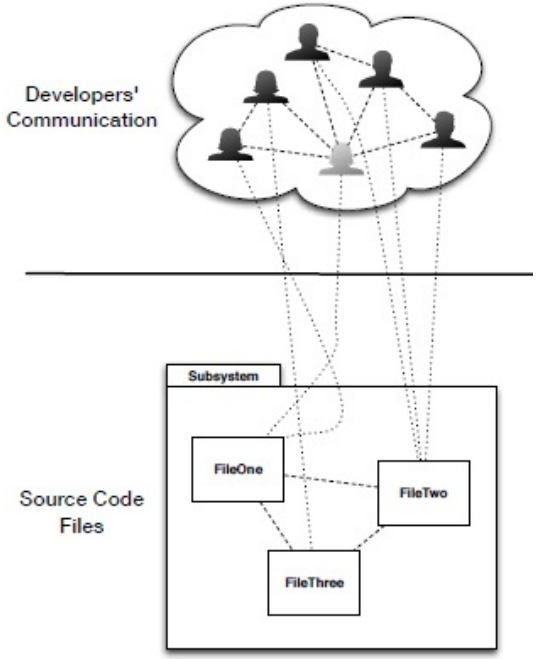
Brooks, in his seminal work *The Mythical Man-Month* [12], noted the scaling issues that arise in large software teams: the number of potential interactions grows quadratically with team size, thus quadrupling when the team size is doubled. Clearly, without organization of some kind, both within the software and the community that develops it, there is a limit to how much projects can be scaled.

In traditional, commercial software projects, the response to the Brooksian critique of large teams is to divide and conquer, *by fiat*. The system is deliberately divided into smaller components, and the developer pool grouped into manageable teams which are then assigned to those components. With well-defined interfaces, the teams' efforts are confined to smaller groups, and the coordination needs are moderated. Software design principles such as separation of concerns [53] play a part in this, as does "Conway's Law" [16], which connects artifact structure with organizational structure.

By contrast, Open Source Software (OSS) projects are not formally organized, and have no pre-assigned command and control structure. No one is forced to work on a particular portion of the project. Team members contribute as they wish in any number of ways: by submitting bug reports, lending technical knowledge, writing documentation, improving the source code in various area of the code base, etc. It has been observed by Sosa *et al.* [56] that the fixed organizational structure found in commercial settings may lead to misalignment with evolving complex products. Henderson and Clark point out that it may may actually hinder innovation [32]. Thus the lack of a rigid organizational structure may in fact be a boon to OSS projects. However, the absence of any structure at all may be just as harmful. Henderson and Clark [32] found that "architectural knowledge tends to become embedded in the structure and information-processing procedures of established organizations". Modularizing artifacts and mapping artifact tasks onto organizational units is a well known solution to the problem of complex product development in organizational management literature [56]. The question then arises, is the social structure of OSS projects free of such constraints and actually unorganized and free-for-all? Do they stand in contrast to the structured, hierarchical style of traditional commercial software efforts? Or, do OSS projects have some latent[1] structure of their own? Are there dynamic, self-organizing subgroups that spontaneously form and evolve?

[1]By latent, we mean not explicitly stated, but observable.

Bird et al. - FSE 2008



How Developers' Collaborations Networks Identified from Different Sources Differ?

IRC CHAT LOG

ISSUE TRACKER

MAILING LIST

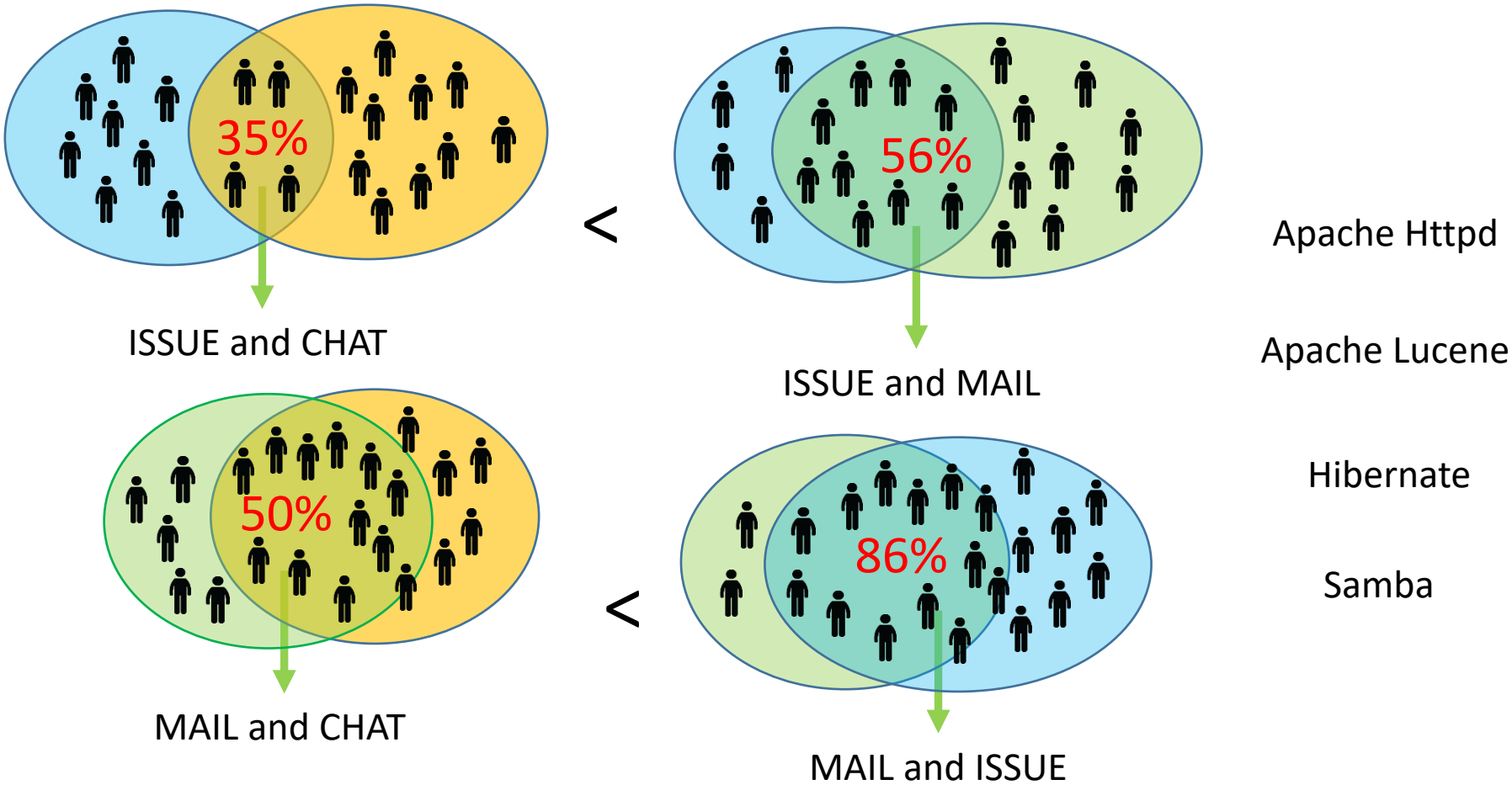
VERSIONING SYSTEM

Sebastiano Panichella, Gabriele Bavota, Massimiliano Di Penta, Gerardo Canfora, Giuliano Antoniol
How Developers' Collaborations Identified from Different Sources Tell us About Code Changes.
The 30th International Conference on Software Maintenance and Evolution (IEEE ICSME 2014)

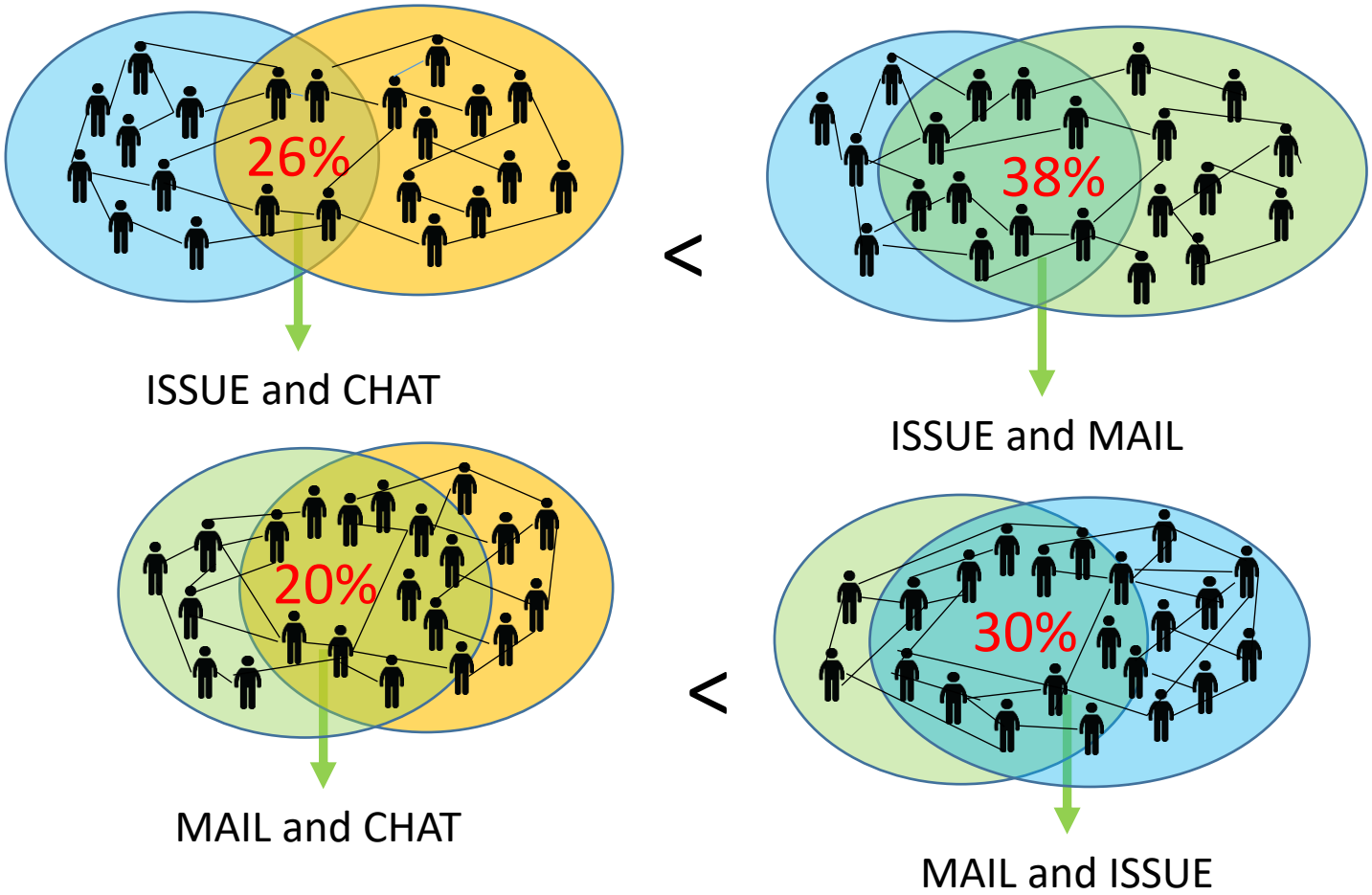
How Developers' Collaborations Networks Identified from Different Sources Differ?

Example: Hibernate OSS Project

Developers Overlap between Different Sources



Overlap of Developers Social Links



Apache Httpd

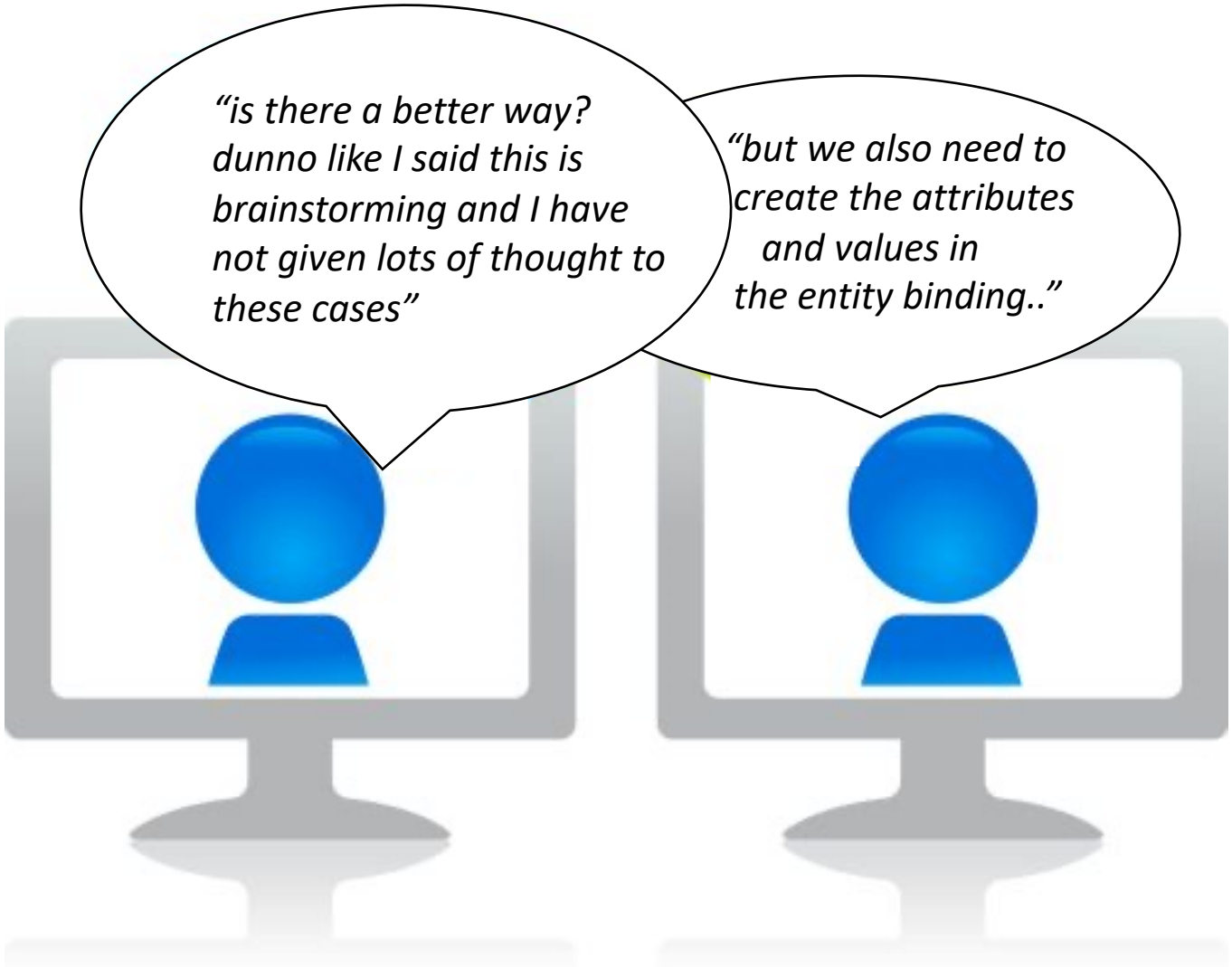
Apache Lucene

Hibernate

Samba

During an IRC Chat Meeting

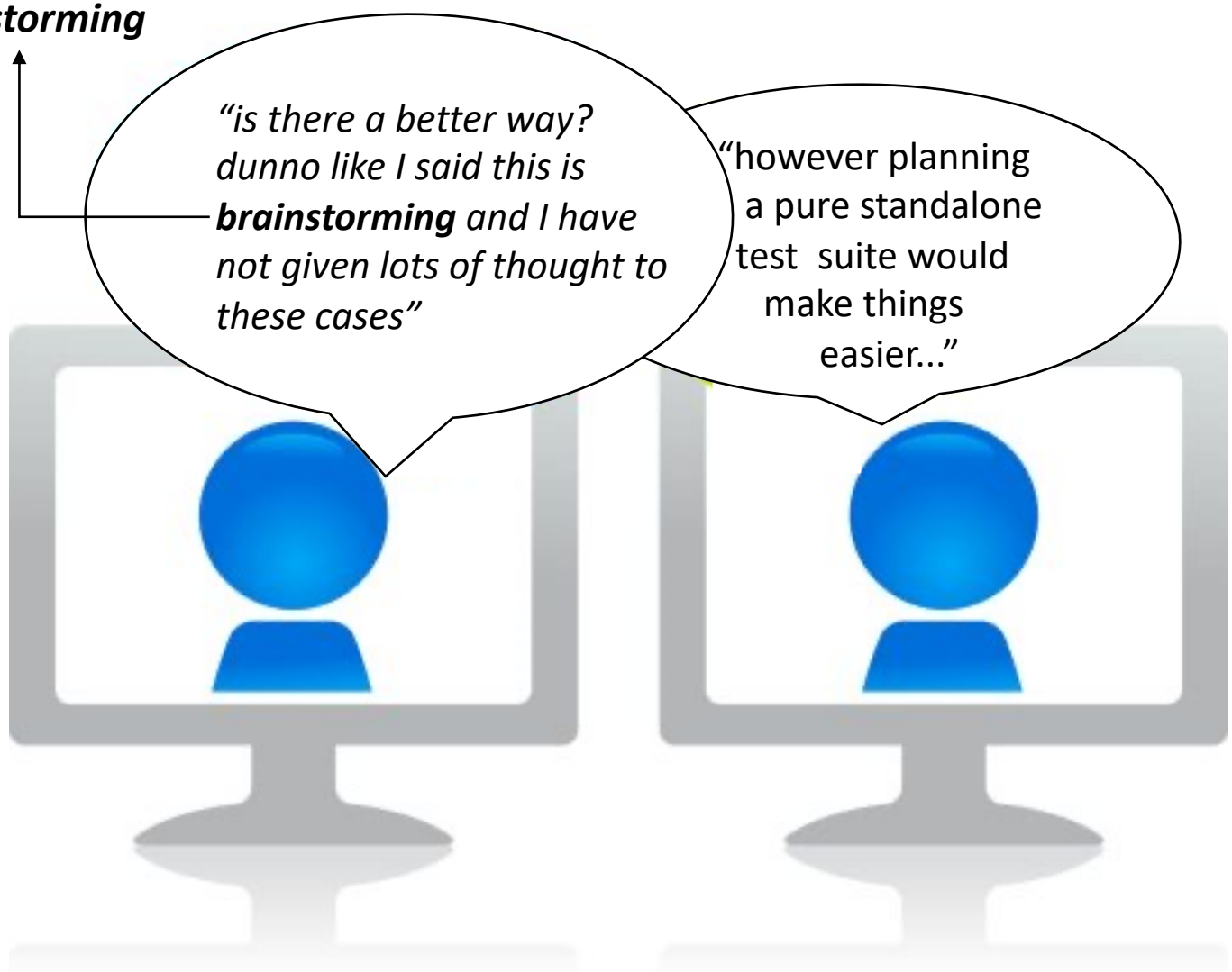
PROJECT: Hibernate



During an IRC Chat Meeting

PROJECT: Hibernate

1) Brainstorming



During an IRC Chat Meeting

PROJECT: Hibernate

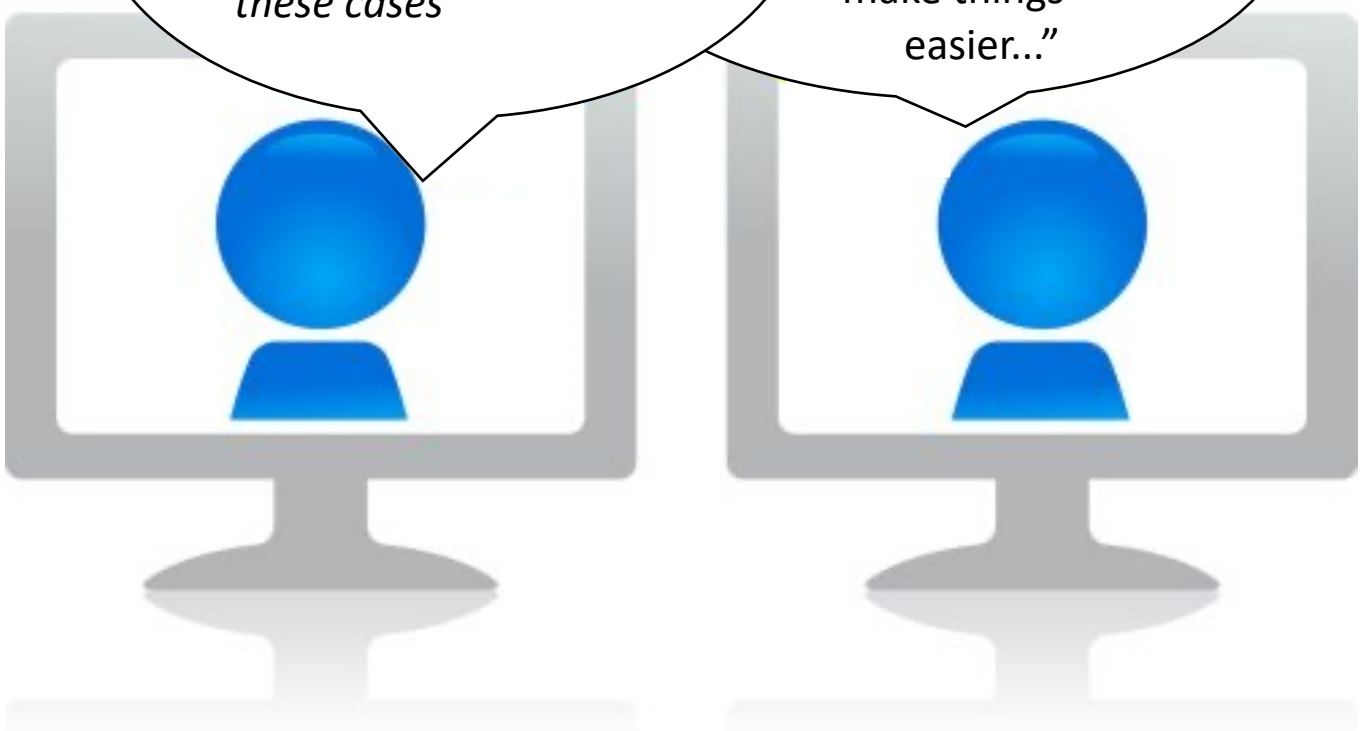
1) Brainstorming

2) Planning

(e.g. Testing activities)

*"is there a better way? dunno like I said this is **brainstorming** and I have not given lots of thought to these cases"*

"however **planning** a pure standalone test suite would make things easier..."



During an IRC Chat Meeting

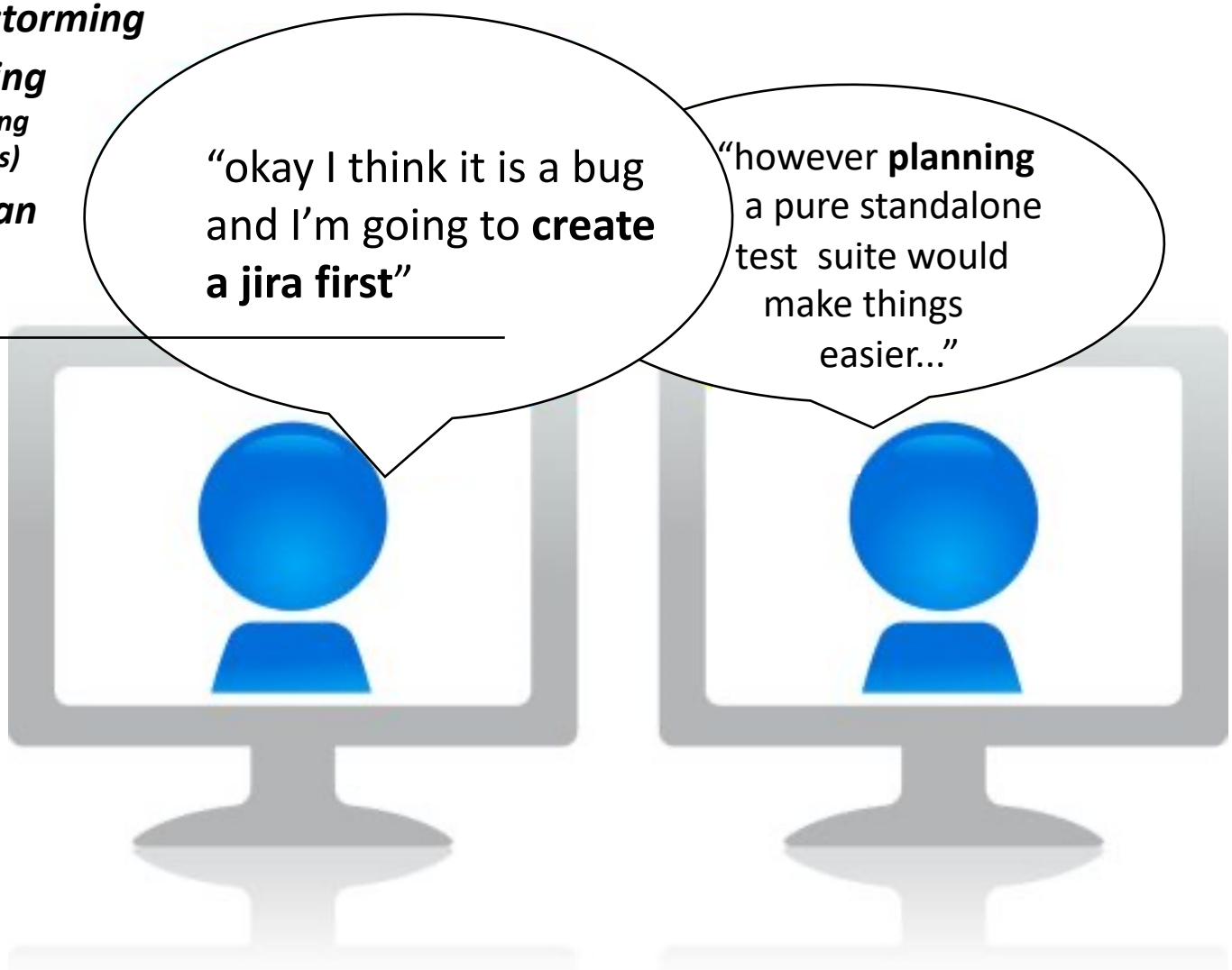
PROJECT: Hibernate

1) Brainstorming

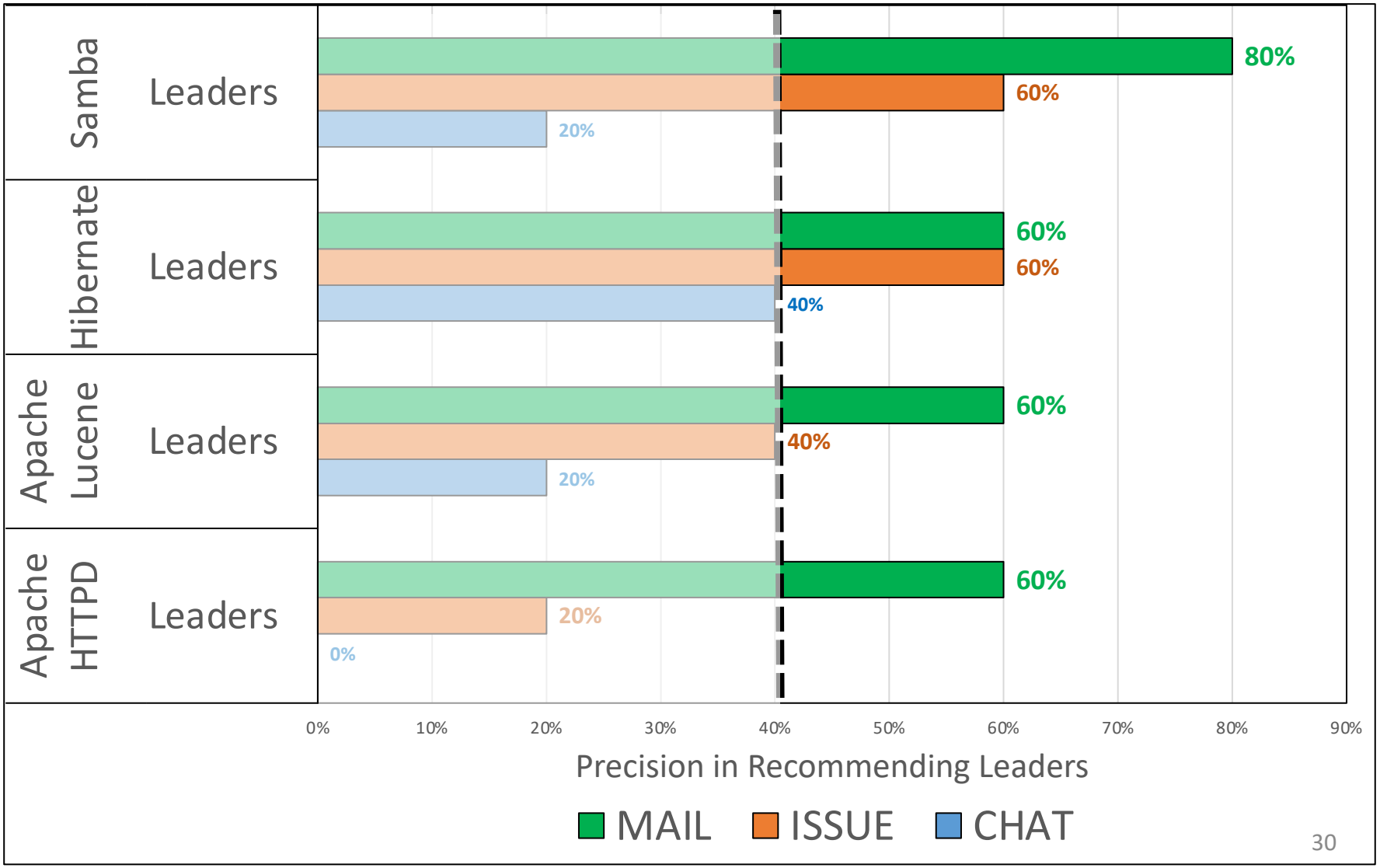
2) Planning

(e.g. Testing activities)

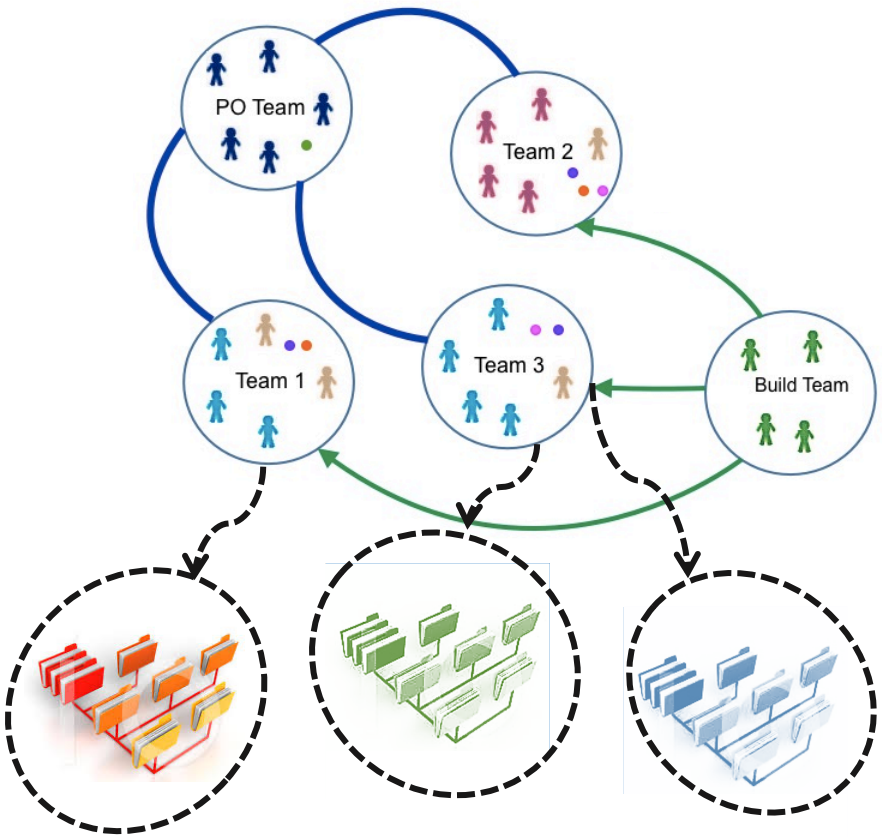
3) Open an Issue



Use Issue, Chat and Mail to Identify Leaders



Analysis of the Evolution of Teams: Why?



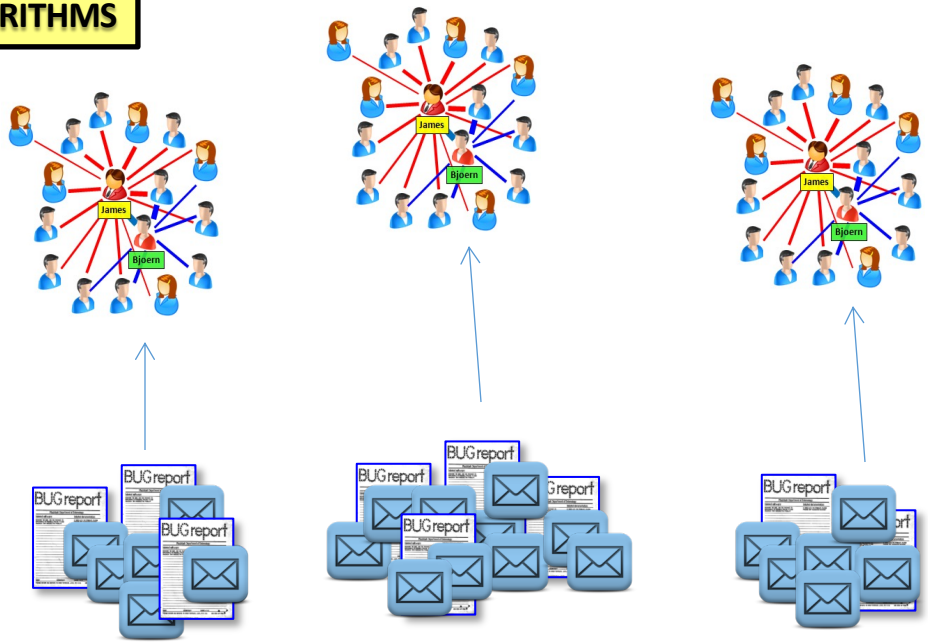
1) To Better Understand the Reasons Behind the Teams Reorganization
(split/merge of developers teams)

2) Investigate whether Emerging Teams Evolve with the aim of Working on more Cohesive Groups of Files. Than Support Re-factoring, Remodulation.

Sebastiano Panichella, Gerardo Canfora, Massimiliano Di Penta, Rocco Oliveto:
How the evolution of emerging collaborations relates to code changes: an empirical study.
The 22nd International Conference on Program Comprehension (IEEE ICPC 2014)

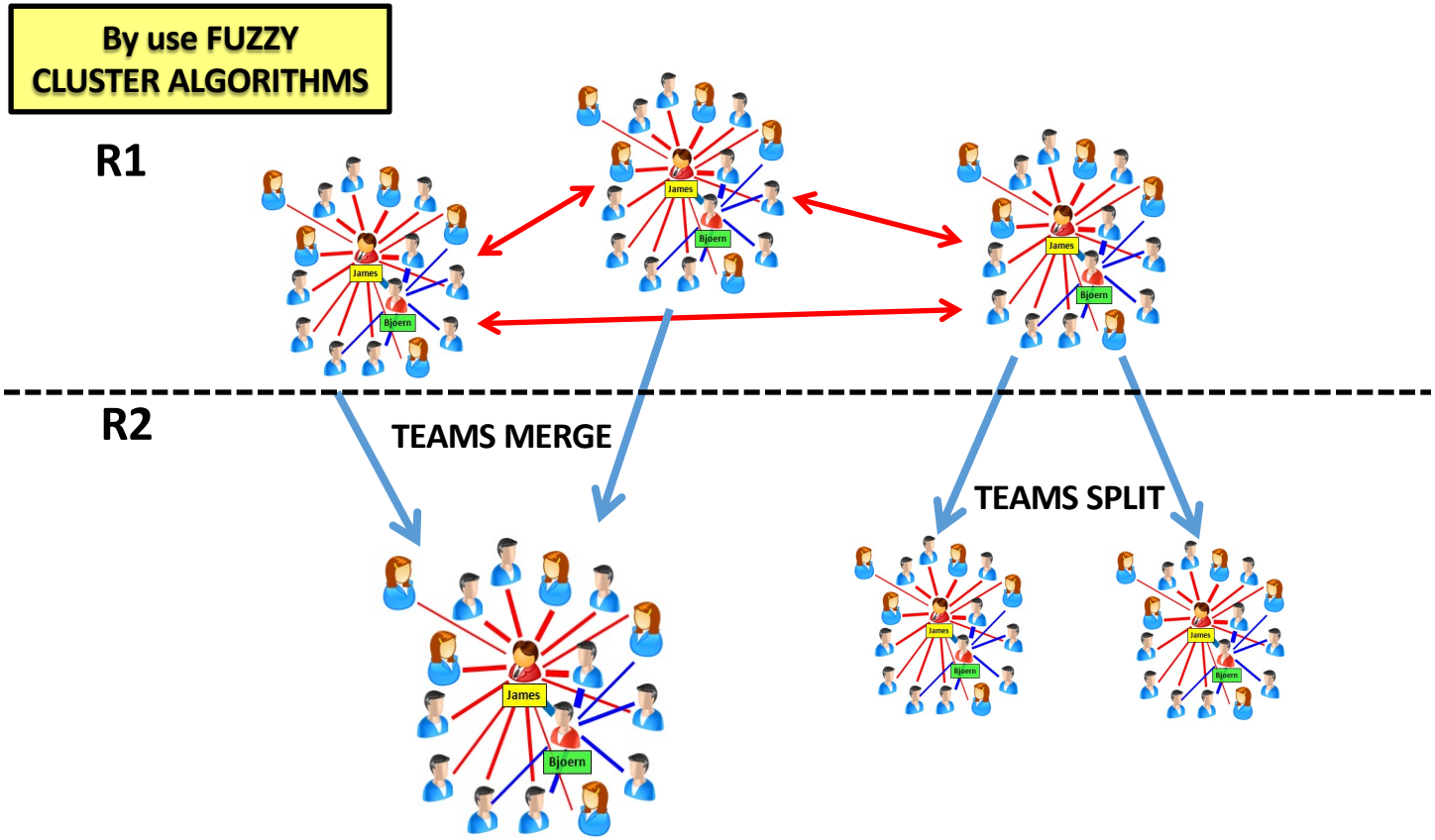
Analysis of the Evolution of Teams: How?

By use FUZZY CLUSTER ALGORITHMS

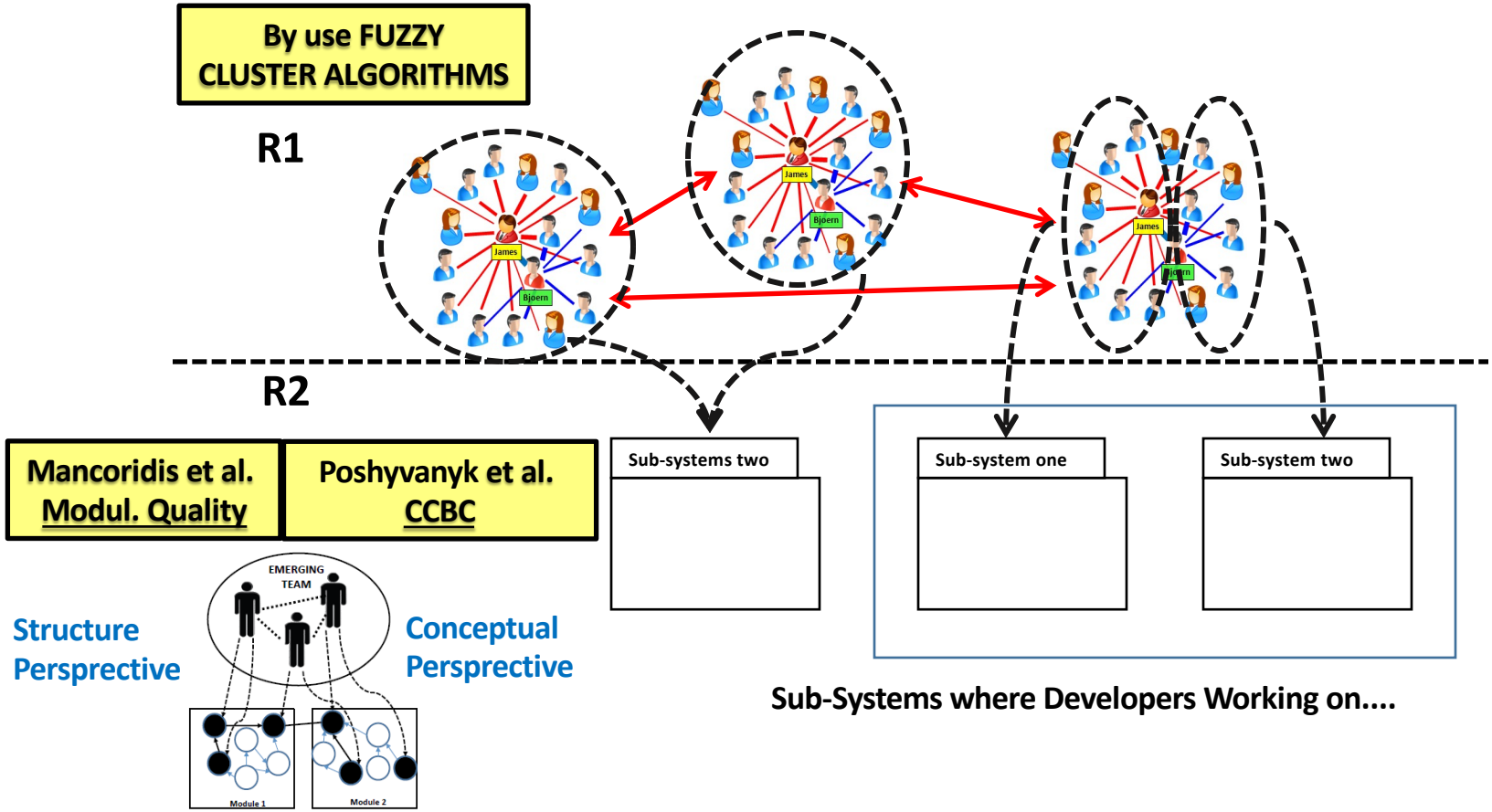


Teams Identification from Emergent Collaborations

Analysis of the Evolution of Teams: How?



Analysis of the Evolution of Teams: How?



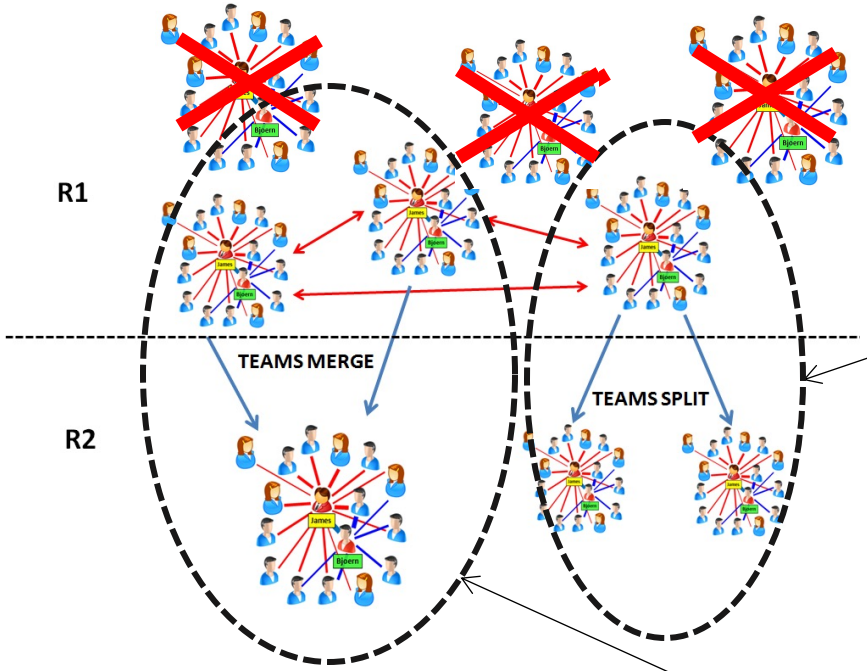
Case Study

- Goal: analyze data from mailing lists/issue trackers and versioning systems
- Purpose: observe the reorganization of the teams between releases
- Quality focus: better understand the reason behind the reorganization of teams

	Apache HTTP	Eclipse JDT	Netbeans	Samba
Period considered	09/1998-03/2012	01/2002-12/2011	01/2001-08/2012	01/2000-09/2011
Releases Considered	2.0 2.2.0 2.2.4 2.2.12 2.4.1	3.0 3.2 3.4 3.6 4.2	3.4 3.6 5.5 6.9 7.2	2.3 3.0.20 3.0.25 3.5.0 4.0

Systems characteristics: Period of Time and Releases Considered

How do Emerging Collaborations Change across Software Releases?



Teams Split in a New Release:
- In 15%-35% of the cases

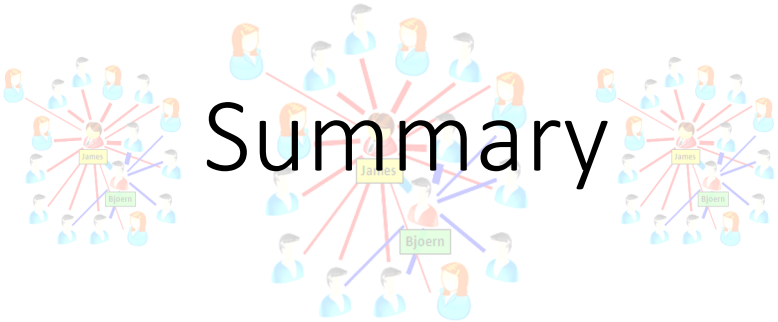
Teams Disappeared:
22%-45%

Teams Survived:
50%-70%

Teams Merge in a New Release:
- in 20%-35% of the cases

PART I

Analysis of Developers' Communication



Summary

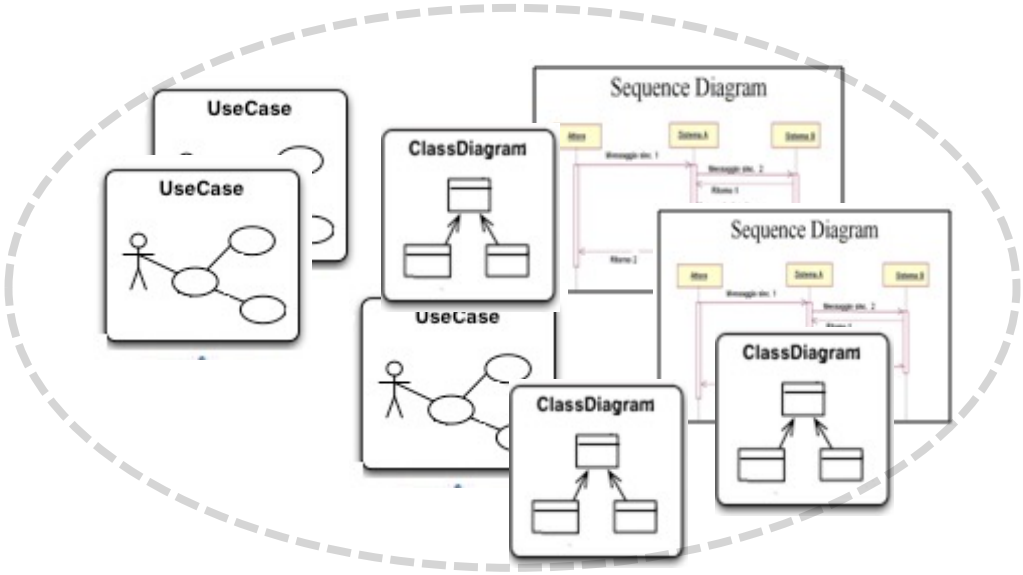
1) Social network recommenders should <u>not limit</u> their information mining a single source.

2) Issue and mail can be used to identify leaders with high accuracy.

3) Social interaction between developers can be used to building better recommenders for software re-modularization or refactoring actions.

PART II

How Developers Browse and Understand Software Artifacts



Two Empirical Studies Aimed at Understanding

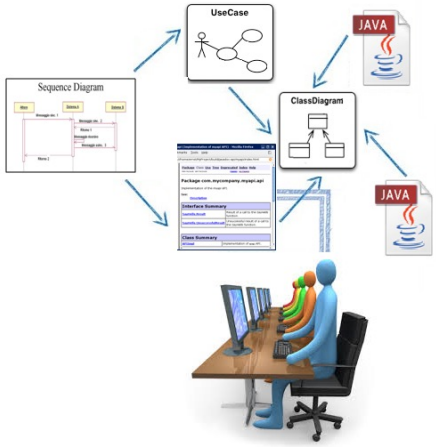
PART II – Experiment A

PART II – Experiment B

Two Empirical Studies Aimed at Understanding

PART II – Experiment A

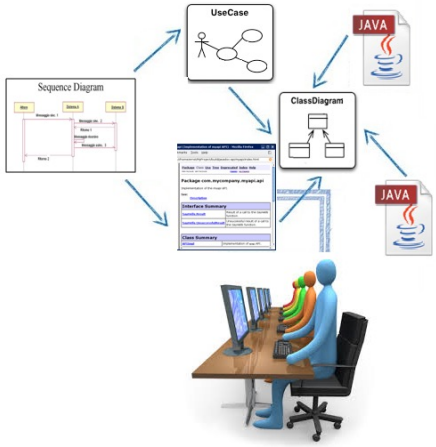
How such documentation is browsed by developers to perform maintenance activities?



PART II – Experiment B

Two Empirical Studies Aimed at Understanding

PART II – Experiment A
How such documentation is browsed by developers to perform maintenance activities?



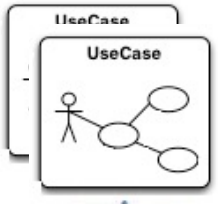
PART II – Experiment B
What code elements are often used by humans when labeling a source code artifact?

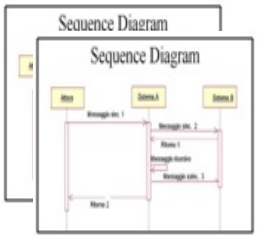
Experiment A: Context

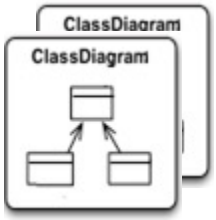
- Object: software artifacts from SMOS, a school automation system developed by graduate students at the University of Salerno (Italy).

121 

121 

67 

72 

6 

- Subjects: 33 participants.

11 Bachelor Students

18 Master Students

4 PhD Students

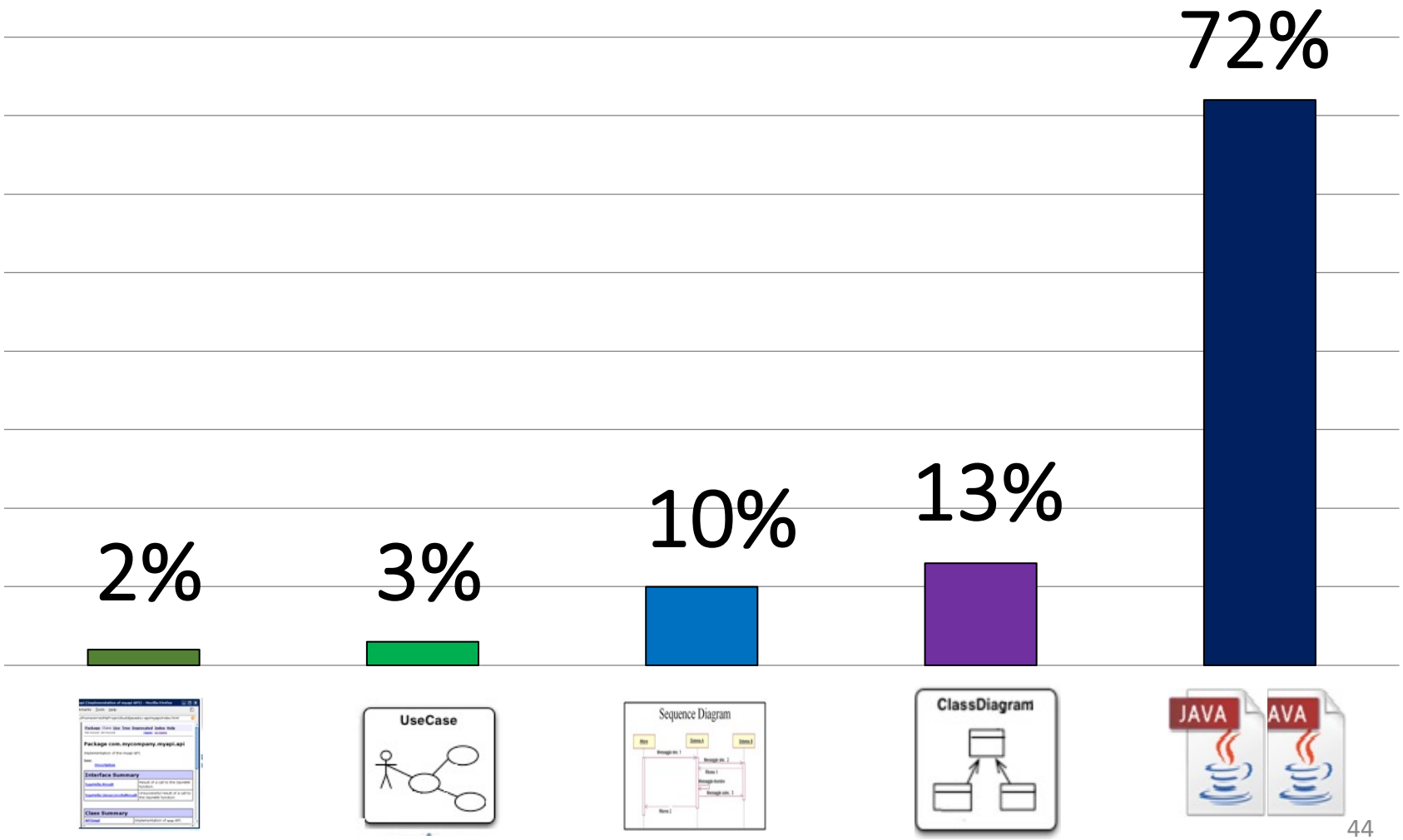
Maintenance Tasks

Bug Fixing:

Add a new feature:

Improve existing features:

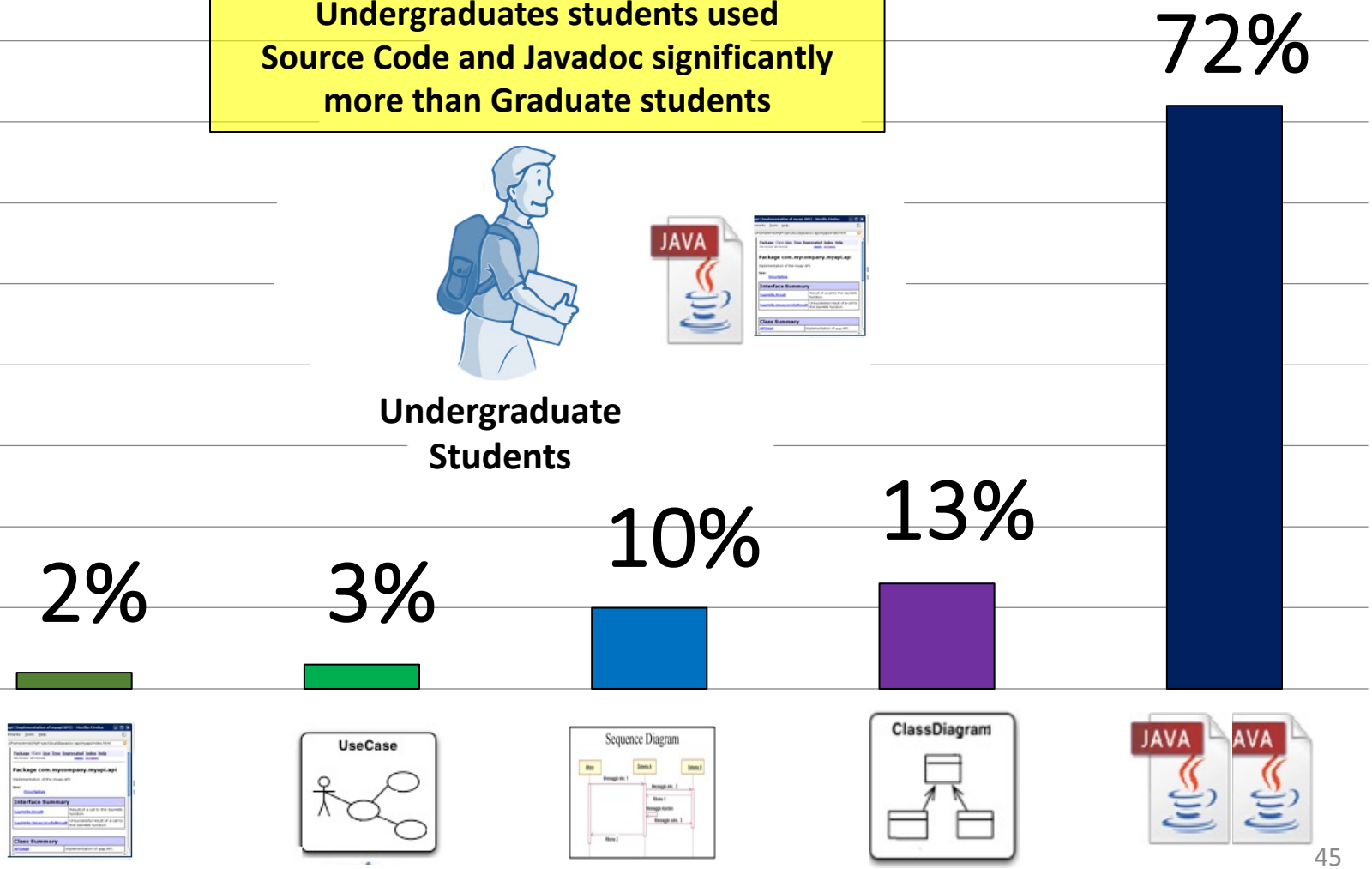
How Much Time did Participants Spend on Different Kinds of Artifacts?



How Much Time did Participants Spend on Different Kinds of Artifacts?

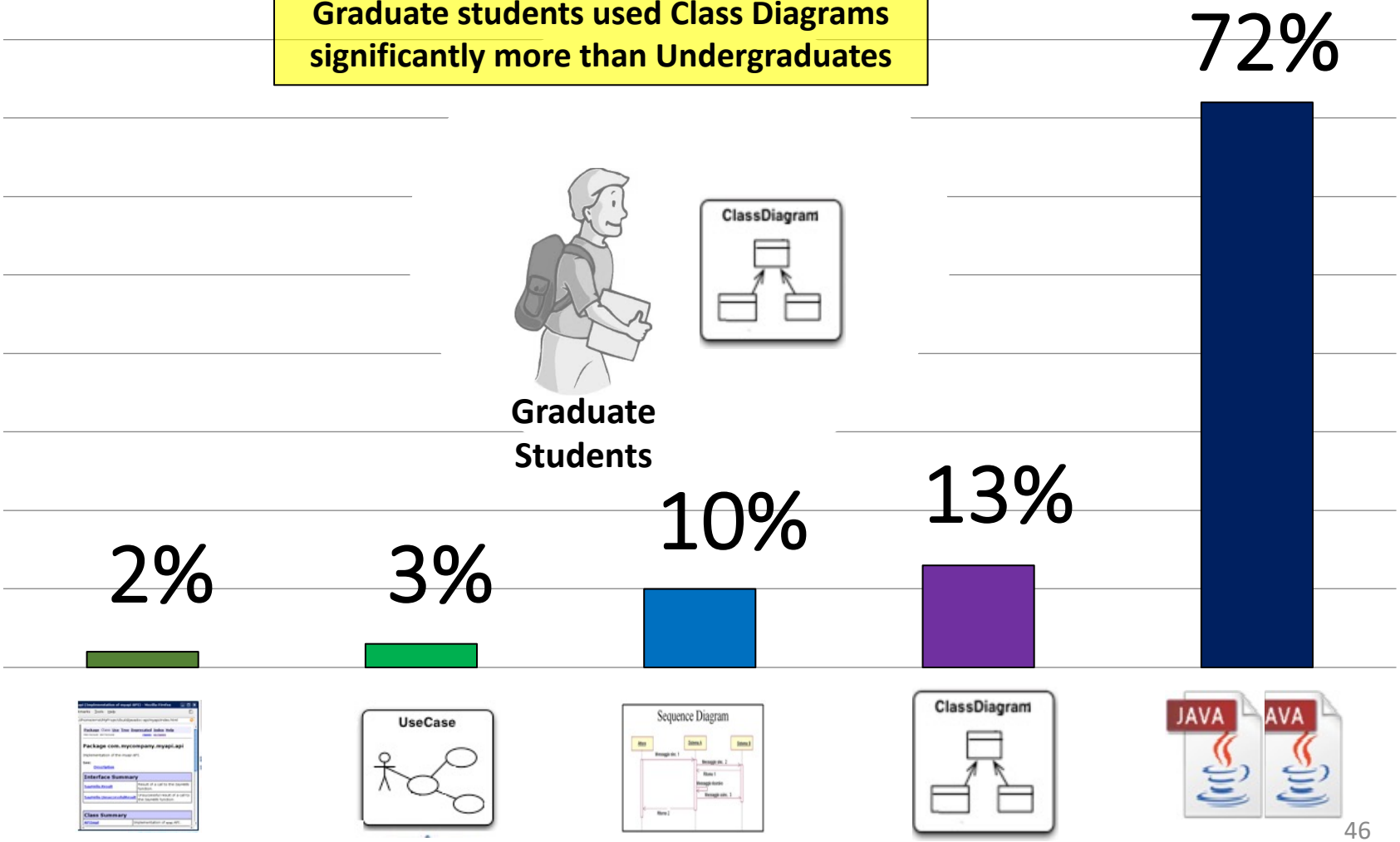
Undergraduates students used Source Code and Javadoc significantly more than Graduate students

Undergraduate Students



How Much Time did Participants Spend on Different Kinds of Artifacts?

Graduate students used Class Diagrams significantly more than Undergraduates



Navigation Patterns Followed By Developers Before Reaching Source Code

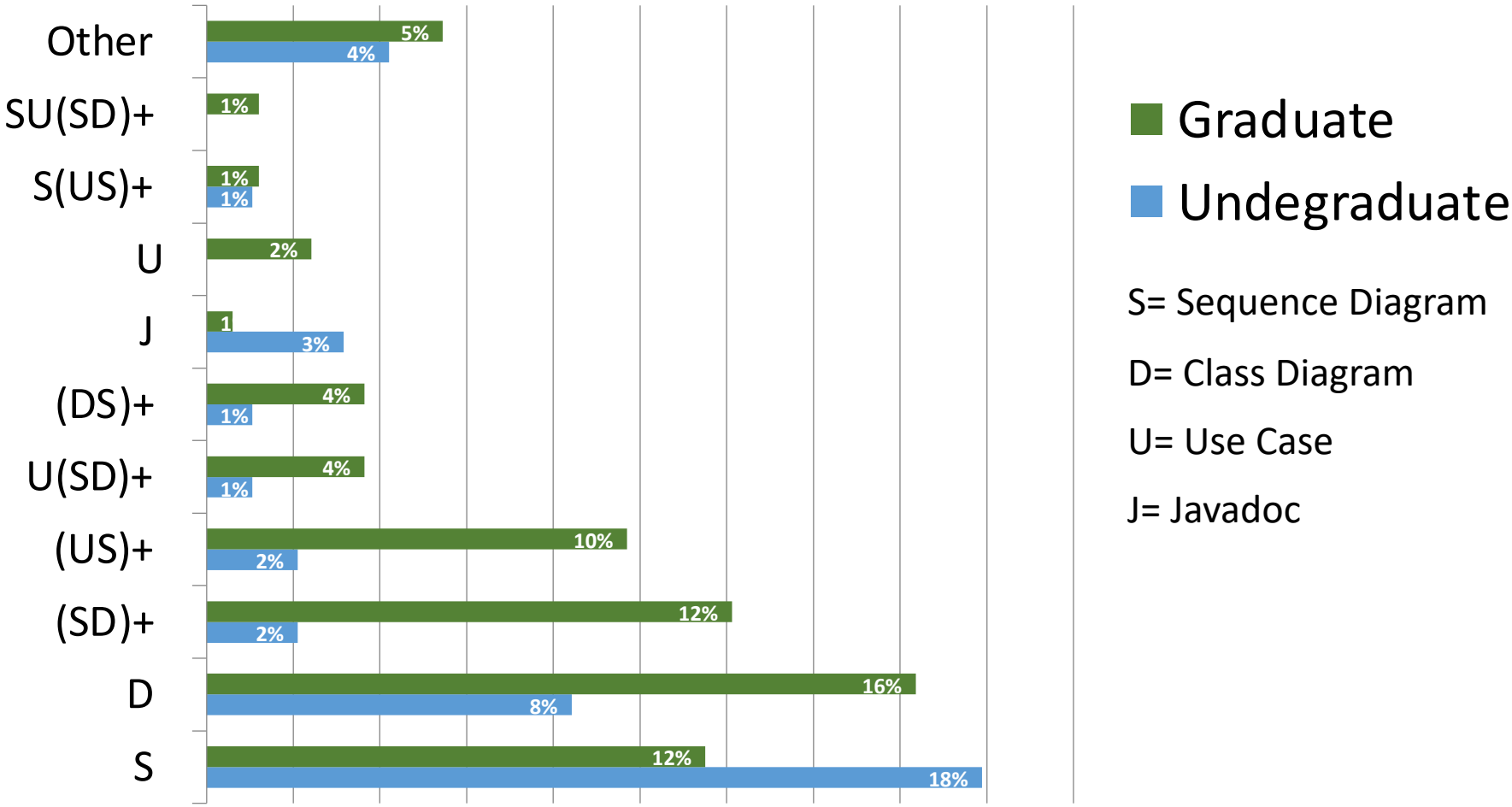
Simple Navigation Patterns

- S = Sequence Diagram
- D = Class Diagram
- U = Use Case
- J = Javadoc

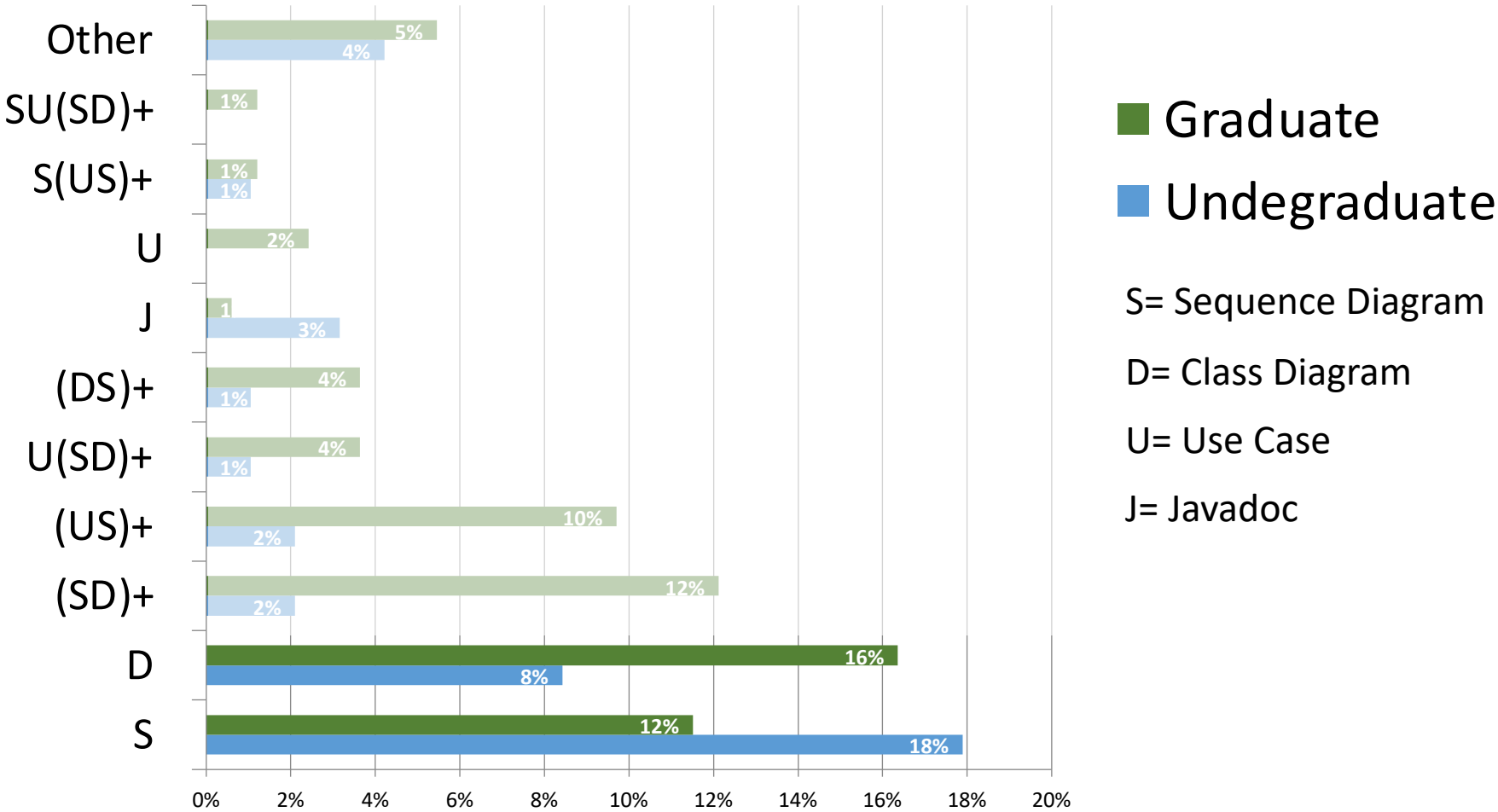
Complex Navigation Patterns

- (SD)+ = Sequence Diagram before Class Diagram
- (US)+ = Use Case before Sequence Diagram
- (DS)+ = Class Diagram before Sequence Diagram
- U(SD)+ = Use Case before (SD)+

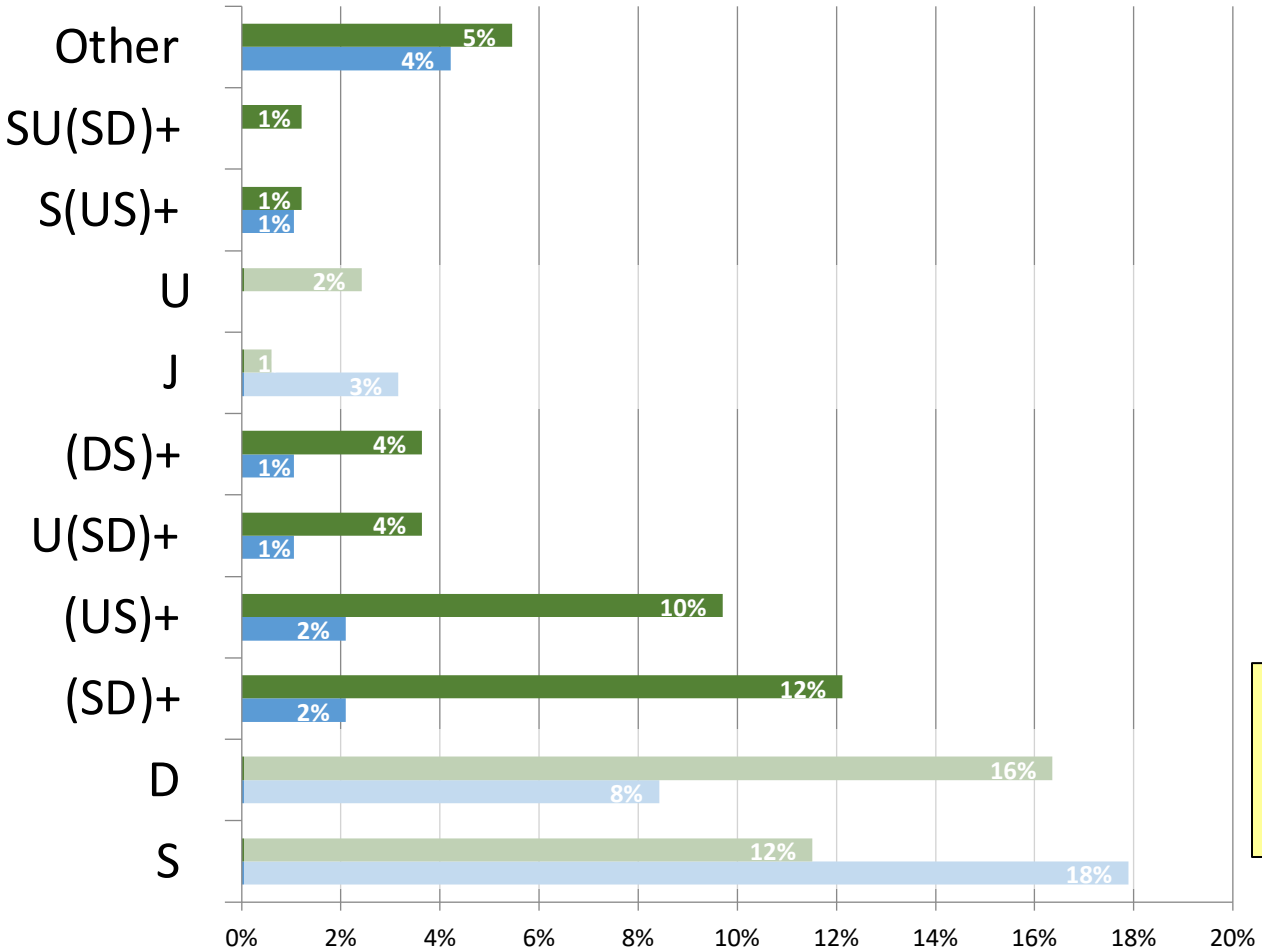
Most Frequent Navigation Patterns Before Reaching Source Code



Most Frequent Navigation Patterns Before Reaching Source Code



Most Frequent Navigation Patterns Before Reaching Source Code



Graduate

Undegraduate

S= Sequence Diagram

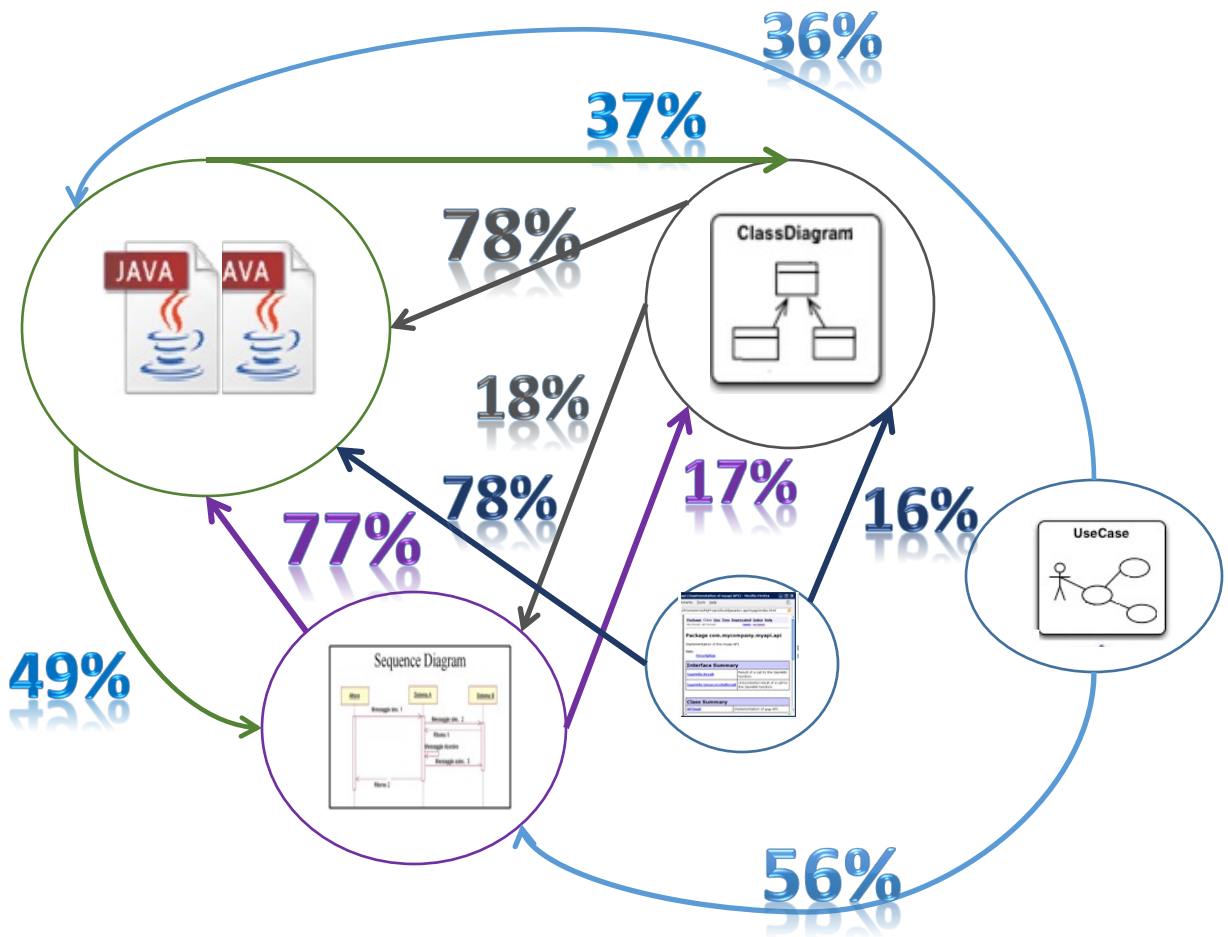
D= Class Diagram

U= Use Case

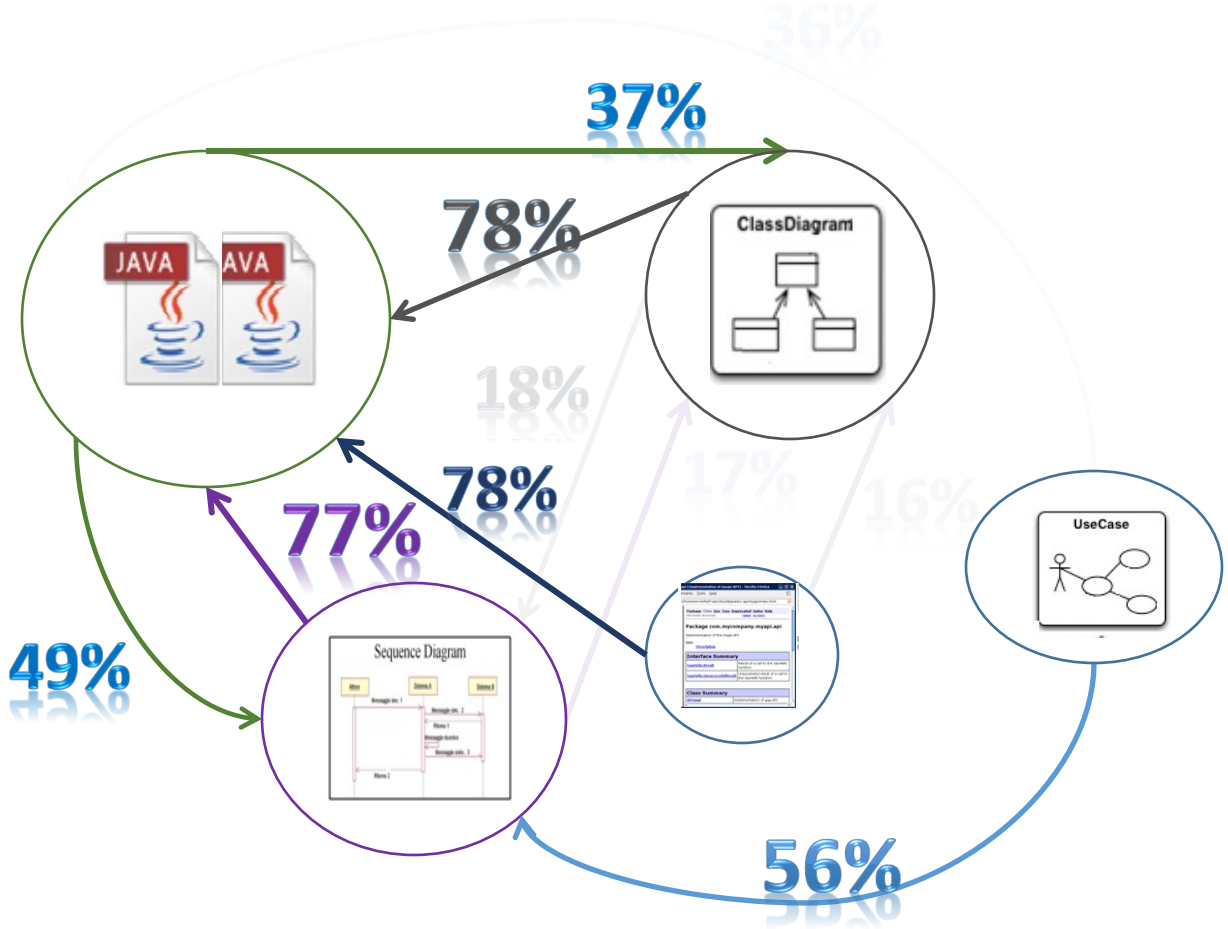
J= Javadoc

More experienced participants use a more "Integrated approach"

Transition Graph between Kinds of Software Artifacts



Transition Graph between Kinds of Software Artifacts



1) From Source Code participants in most cases "go back" to Sequence and Class Diagrams

2) From Sequence and Class Diagrams participants in most cases "go back" to Source Code

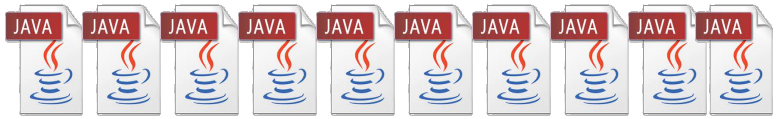
3) Starting from a Use Case, participants go ahead reading Sequence Diagrams. Only after, they reading and writing Source Code

PART II – Experiment B

What Code Elements are Often Used by Humans When Labeling a Source Code Artifact?

Experiment B: Context

- Object:



eXVantage (industrial test data generation tool)

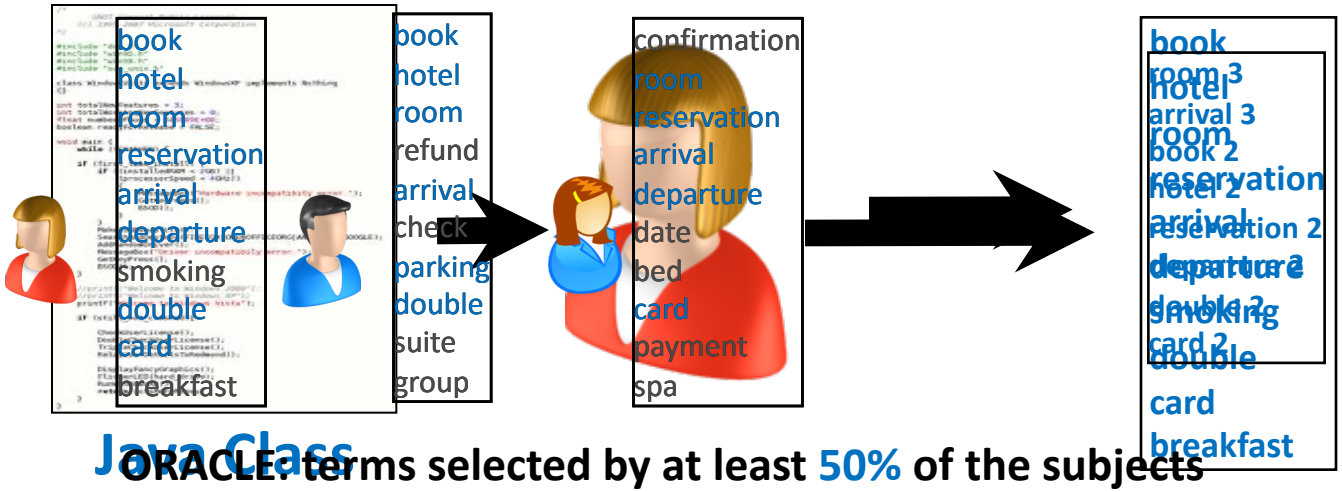
- Subjects:

17 Bachelor Student CS

(Univ. of Molise, second year)

21 Master Student in CS

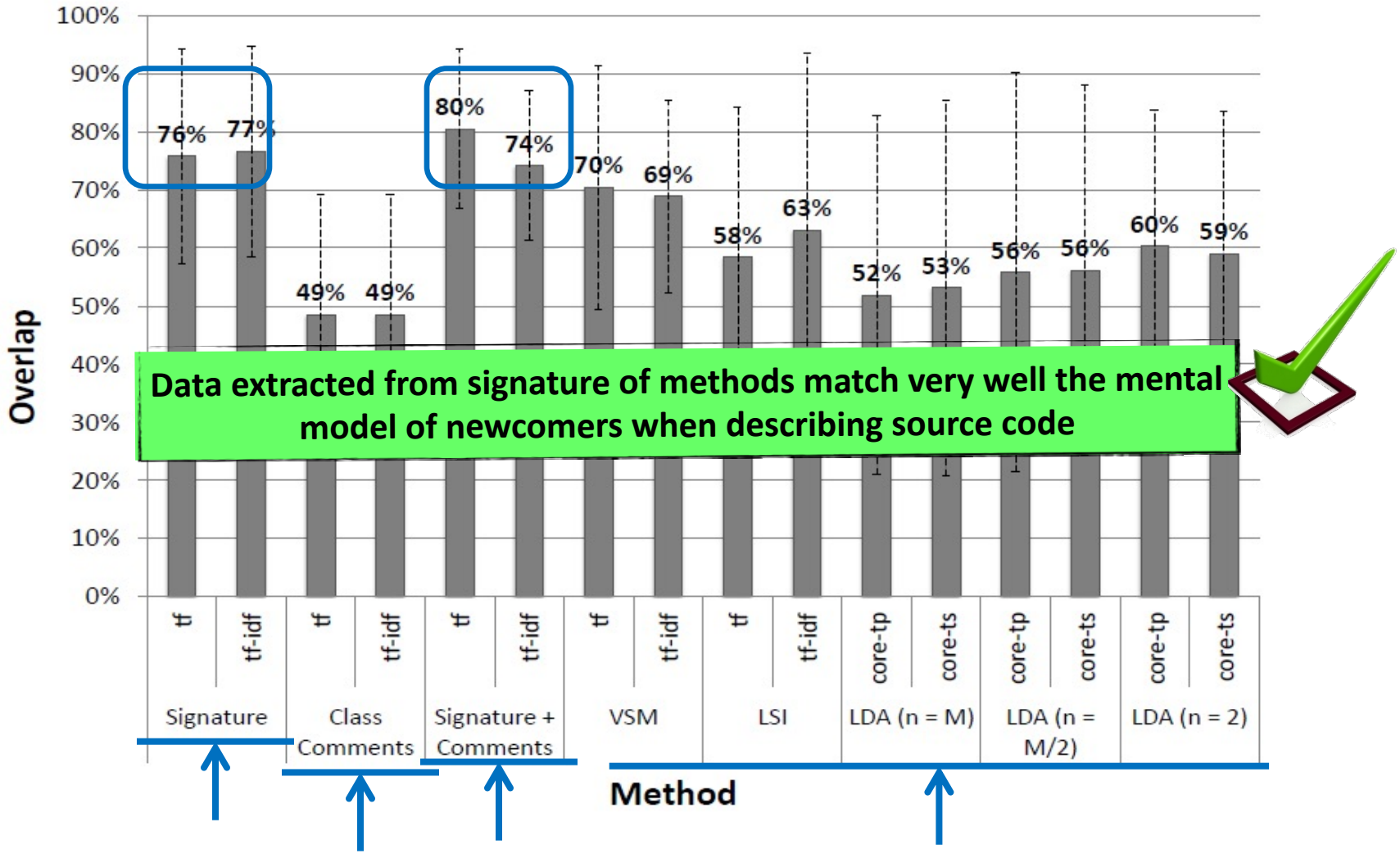
(University of Salerno)



Java Class

ORACLE: terms selected by at least 50% of the subjects

Comparison of Different Labeling Techniques

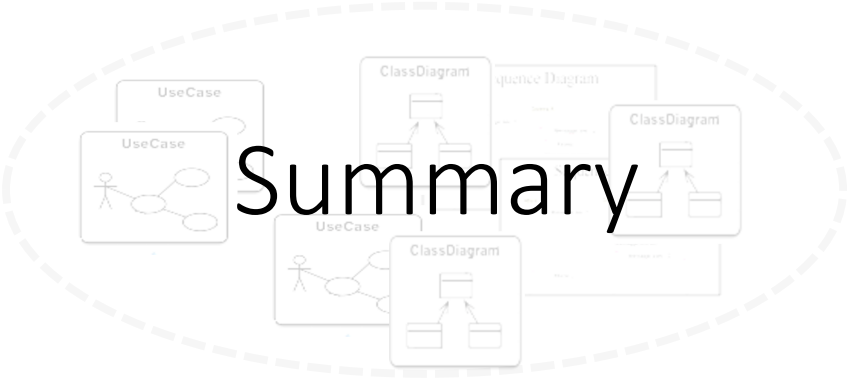


Data extracted from signature of methods match very well the mental model of newcomers when describing source code

Andrea De Lucia, Massimiliano Di Penta, Rocco Oliveto, Annibale Panichella, Sebastiano Panichella : Labeling Source Code with Information Retrieval Methods: An Empirical Study. EMSE 2014.

PART II

How Developers Browse and Understand Software Artifacts



Summary

1) **Newcomers spend more time to analyze low-level artifacts as compared to high-level artifacts**

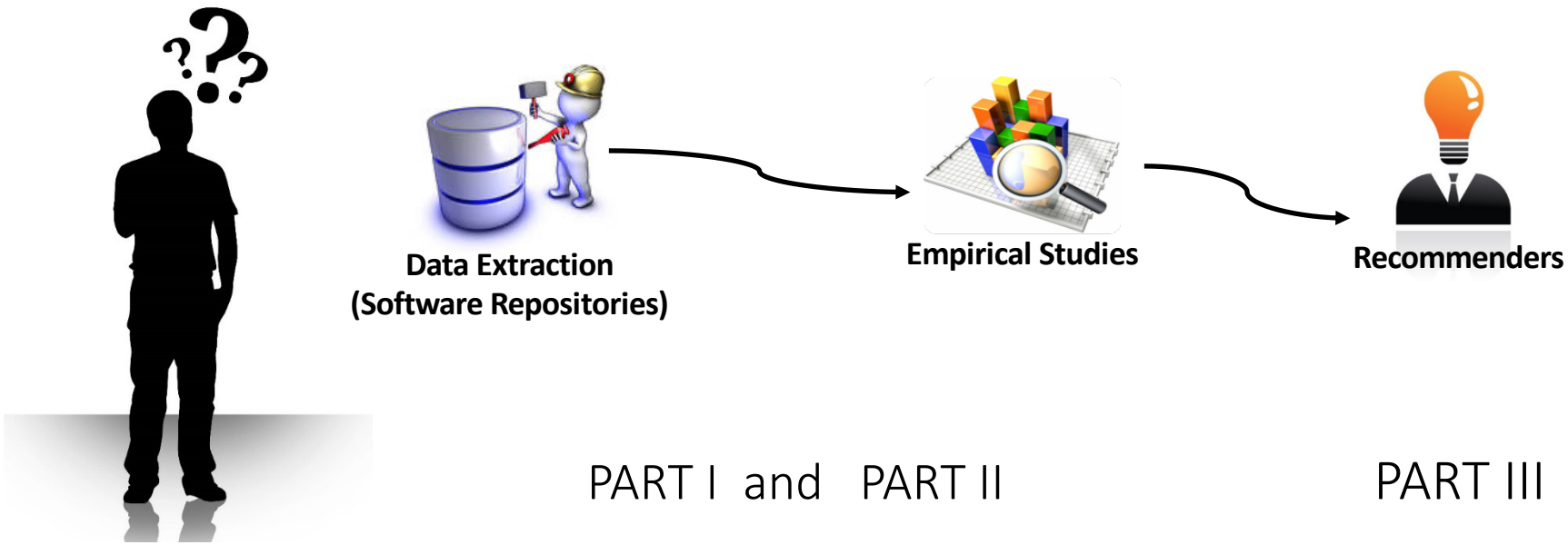
2) **Less experienced newcomers spend a significantly higher proportion of time on source code**

3) **More experienced newcomers, instead, spend more time on class diagrams**

4) **Heuristics based on data extracted form signature of methods are able to match very well the mental model of newcomers when describing source code elements**

PART III

Recommenders



PART I and PART II

PART III

Two Recommenders to Support Project Newcomers

PART III - A)
Suggest Appropriate Mentors to Help Newcomers in Open Source Projects

PART III – B)
Mining Source Code Descriptions from Developers' Communication to Improve Newcomers' Program Comprehension

PART III – A)

Suggest Appropriate Mentors to Help Newcomers in Open Source Projects

Previous Work

Moving into a New Software Project Landscape

Barthélémy Dagenais[1]; Harold Ossher[2], Rachel K. E. Bellamy[2], Martin P. Robillard[1], Jacqueline P. de Vries[2]

School of Computer Science[1]
McGill University
Montréal, QC, Canada
{bart,martin}@cs.mcgill.ca

IBM T.J. Watson Research Center[2]
P.O. Box 704
Yorktown Heights, NY 10598
{ossher,rachel,devries}@us.ibm.com

ABSTRACT

When developers join a software development project, they find themselves in a *project landscape*, and they must become familiar with the various landscape features. To better understand the nature of project landscapes and the integration process, with a view to improving the experience of both newcomers and the people responsible for orienting them, we performed a grounded theory study with 18 newcomers across 18 projects. We identified the main features that characterize a project landscape, together with key orientation aids and obstacles, and we theorize that there are three primary factors that impact the integration experience of newcomers: early experimentation, internalizing structures and cultures, and progress validation.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management

General Terms

Human Factors

1. INTRODUCTION

Software developers working on a project effectively inhabit a *project landscape*. They are familiar with its features, such as the product architecture, the team communication strategies and the development process, and they know the shortcuts and the commonly-traveled paths. Newcomers are explorers who must orient themselves within an unfamiliar landscape. As they gain experience, they eventually settle in and create their own places within the landscape. Like explorers of the natural landscape, they encounter many obstacles, such as a culture shock or getting lost without help.

We conducted a qualitative study to better understand what project landscapes look like and how newcomers explore them. Thinking of a project as a landscape, and integration of newcomers as the process of settling into that landscape, changes what we perceive to be important and helps us see new ways of aiding newcomers. From a newcomer's perspective, it emphasizes the process of learning about a project, and how that process unfolds over time. From the perspective of someone helping newcomers settle in, the landscape metaphor reveals the need to show them the commonly-traversed routes, to help them learn to interpret aspects of the landscape unique to the project, and to introduce them to the customs of the people who inhabit the landscape. It also suggests that if the community wants to be welcoming to newcomers, they need to be tolerant of cultural faux-pas, be sensitive to missteps caused by a newcomer's lack of understanding, take the time to understand why newcomers get lost in their landscape, and add readily-interpretable signposts. Such signposts are especially important at cross-roads, i.e., places with choices where others have tended to get lost. Identifying what counts as a cross-roads and what characterizes the parts of a project that need signposts can be aided by studies such as that presented here.

Specifically, we were interested in answering three main research questions: what are the key, prominent features in a project landscape, what orientation obstacles do new team members face, and what orientation help can be provided? We interviewed 18 developers and team leaders across 18 projects at IBM during the last year to answer these questions.

Following these interviews, we theorized that there are three main factors that impact how newcomers settle into a project landscape: *early experimentation, internalizing structures and cultures*, and *progress validation*. We also identified the landscape features that newcomers learned while moving into new project landscapes and we observed how the features facilitated or hindered the newcomers' integration. When we presented the results of our study to seven of the participants, they all agreed that the factors accurately represented their experiences as newcomers and that application of our findings would have eased their integration.

In the past, studies on project integration have been performed with new employees joining their first software development projects [2, 15]. Because these studies were performed with junior and recently-hired developers, many of the difficulties they encountered related to the newness of the corporate culture and the difference between academic and industrial environments. We were interested in understanding specifically the project landscape, independently of the circumstances related to the first-time transition of personnel into an industry environment. To this end, we focused this study on developers with varying degrees of experience in the field and within their company who were joining on-going projects in the company. We reported preliminary results at a workshop [6].

The contributions of this paper include a theory, grounded in empirical data, of how newcomers integrate into a project landscape, and a characterization of project landscapes as seen by newcomers.

We begin by summarizing the method we used to perform this study, in Section 2. We characterize project landscapes by present-

[1] This research was conducted while the author was working at the IBM T.J. Watson Research Center.

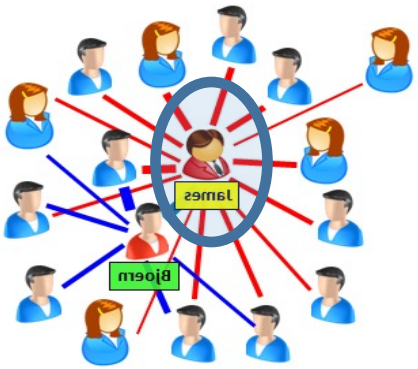
ICSE 2010 Cape Town, South Africa

Mentoring of Project Newcomers is Highly Desirable....

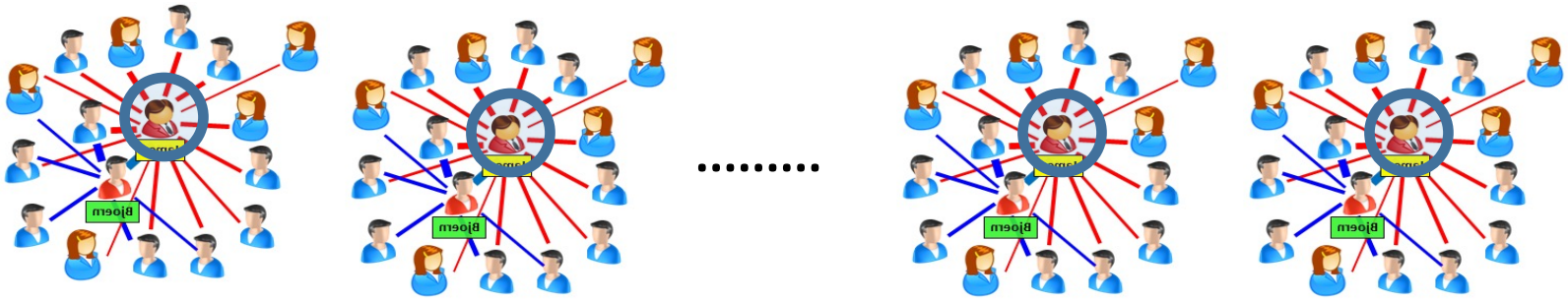
Dagenais et al. ICSE 2010

When a Newcomer Joins a Project

- Small Projects: find Mentors is a trivial problem



- Large Projects: find Mentors is not a trivial problem



Identifying Mentors in Software Projects

Home GSoC General Information For Contributors For Committers/PMCs Mentoring Speakers Calendars Mailing List

The Apache Software Foundation

Google™ Custom Search

Meritocracy in Action.

Home / MentoringProgramme

Many projects in the ASF are able to provide mentors for newcomers. In fact, most projects are happy to assist newcomers to their projects as part of their normal operations. However, some people are looking for more structure. The Mentor Programme of the The Apache Software Foundation provides additional support and structure for people looking to make an initial contribution to an ASF project.

The mentoring programme is not here to teach you to write documentation or code. It is here to help you understand how to make a valuable contribution to an Apache project. You can expect to be guided through our contribution processes. You can also expect to get technical support with respect to your chosen project. You cannot expect your mentor to be a "teacher", they will provide enough information for you to progress within the project. You need to bring the confidence to take their guidance and discover the detail for yourself.

This page is a description of the Mentoring Programme. The program is open for business, but, like many other things at the Foundation, it under constant improvement and revision. Therefore, the description below is marked 'draft.'

Quick Definitions

The ASF believes that the best way for people (and, indeed, entire projects) to join the community is with the help of committed members of the community. A community member who makes a commitment to help a new contributor get started is a *mentor*. The new person, on the other hand, is a *mentee*. Believe it or not, that is the word in the dictionary for this role.

The Foundation is organized into a series of Top Level Projects, or TLPs. The document uses 'TLP' when it is referring to an ASF project. It uses the word 'project' to refer to a the work a mentee does under the Mentoring Programme.

Who can be a mentee?

The Mentor Programme is intended to assist people in becoming contributors to ASF projects. Thus, anyone interested in contributing effort to an ASF project is a potential mentee. You need to be a self starter, your mentor will not take responsibility for "managing" your work here. Everyone who contributes to an Apache project does so on a voluntary basis, there are no managers here - only helpful peers.

Mentoring is a significant volunteer effort, over and above what the mentor is already doing for the project. Therefore, the programme asks mentees to make a material commitment of time to the process. There are no legally binding commitments involved, but a mentee must, as described below, submit a plan for a significant effort and show ongoing progress.

It is important to reiterate that all work on ASF projects is on a volunteer basis. The Foundation does not pay anyone to mentor or contribute.

Applying for the Mentor Programme

There are two simple steps to apply:

1. Review the content below to learn about the details of the requirements.
2. Fill out the application form

https://community.apache.org/mentoringprogramme.html

Characteristics of a Good Mentor

Enough expertise
about the topic of interest
for the newcomer.

Enough ability to help
other people.

YODA
(Young and newcOmer Developer Assistant)

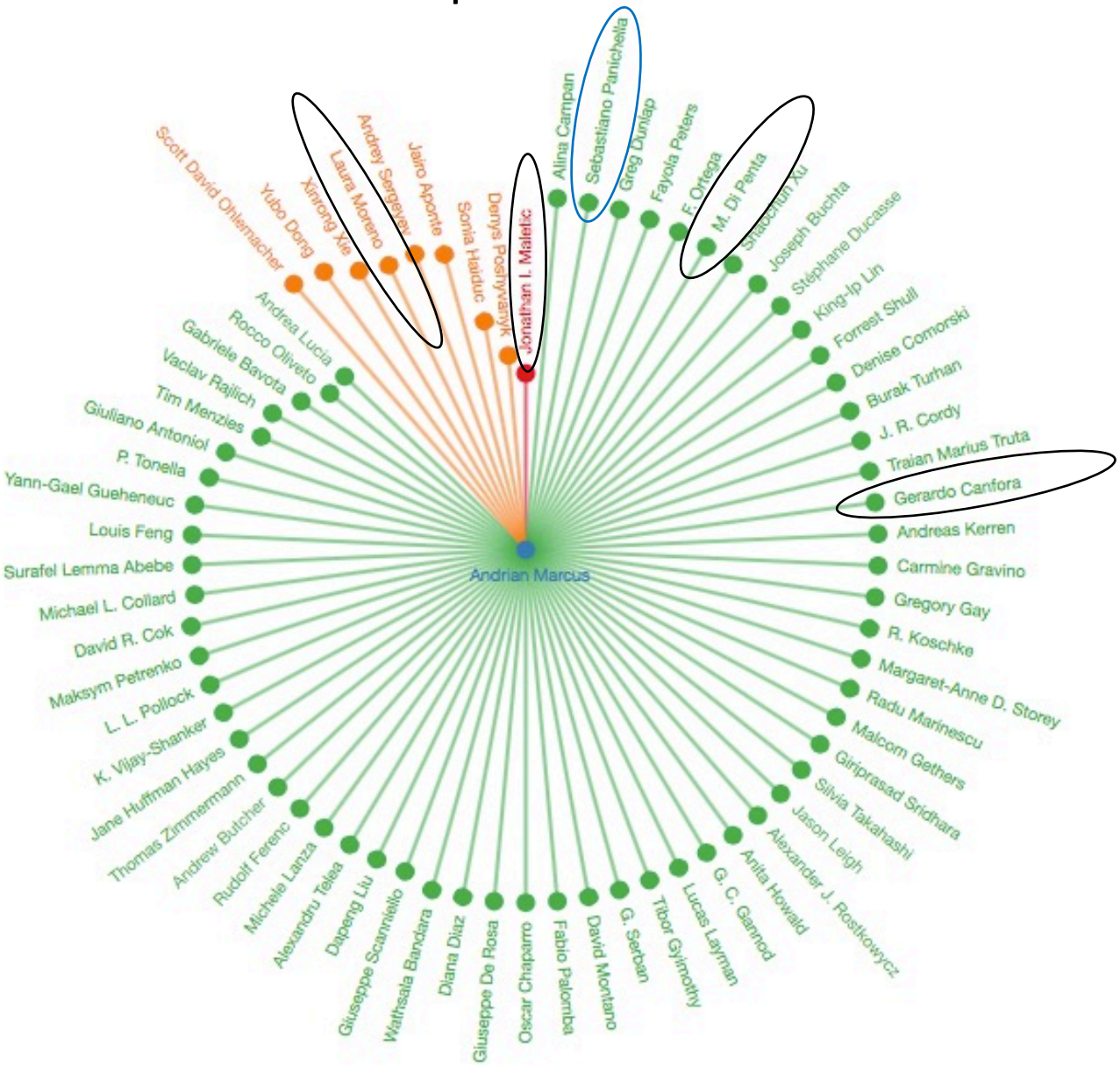
Approach for Mentors Identification in Open Source Projects

1) Find Past Successful Mentors

2) Suggest Mentors Having Specific Skills

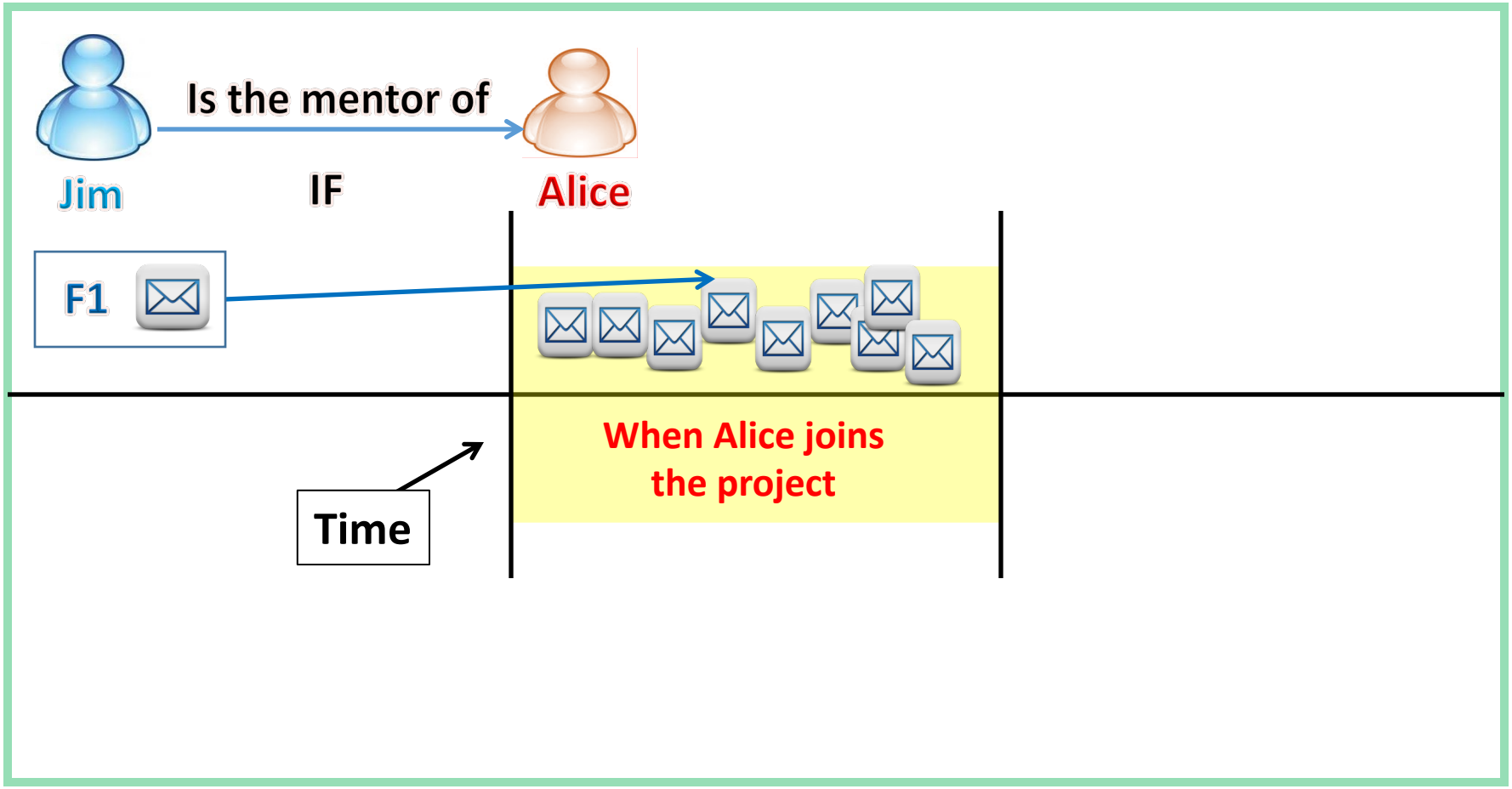
Gerardo Canfora, Massimiliano Di Penta, Rocco Oliveto, Sebastiano Panichella:
Who is Going to Mentor Newcomers in Open Source Projects?
International Symposium on the Foundations of Software Engineering (SIGSOFT FSE 2012)

Source of Inspiration: Arnetminer



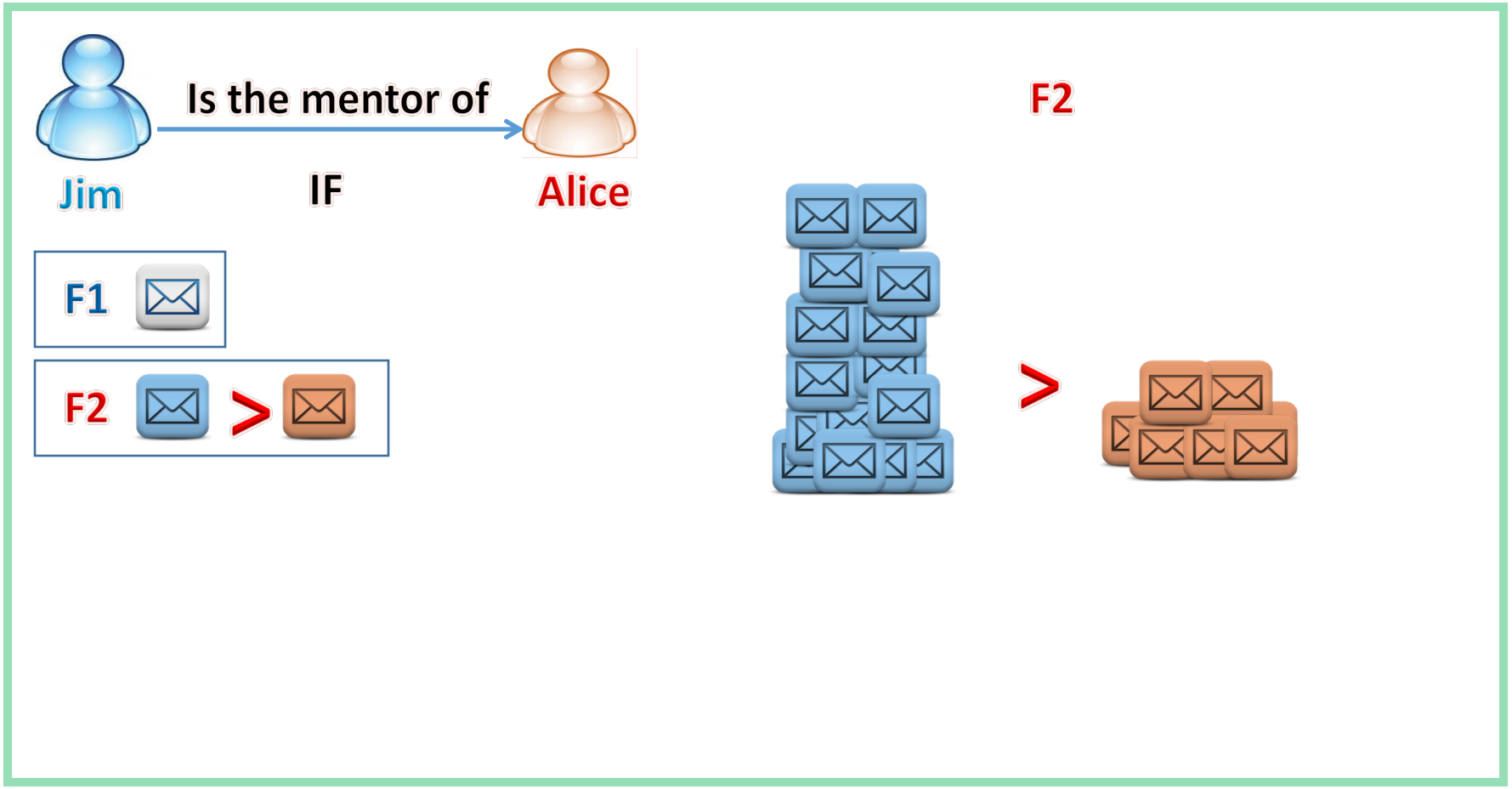
YODA

F1: Exchanged emails



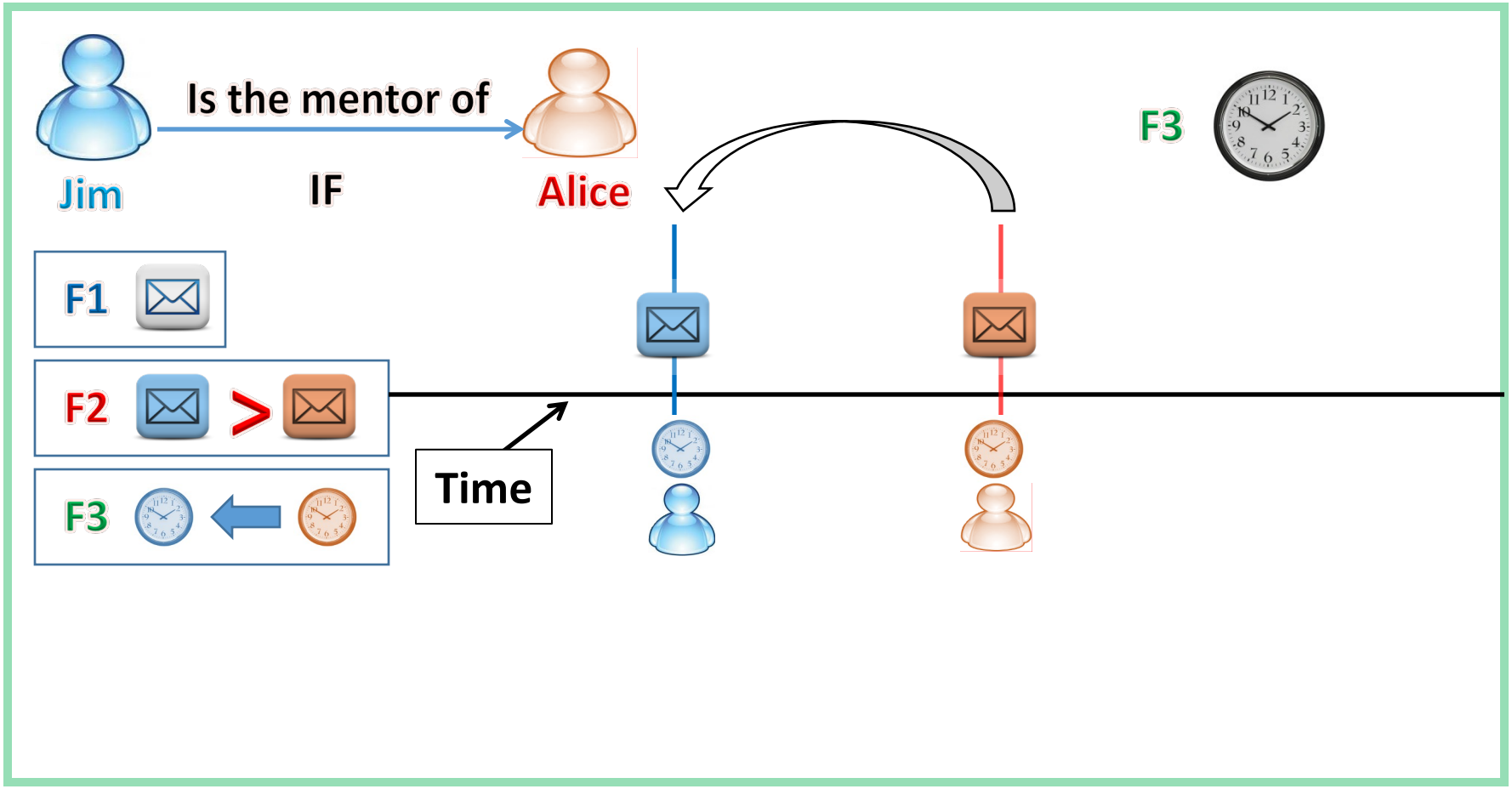
YODA

F2: amount of emails



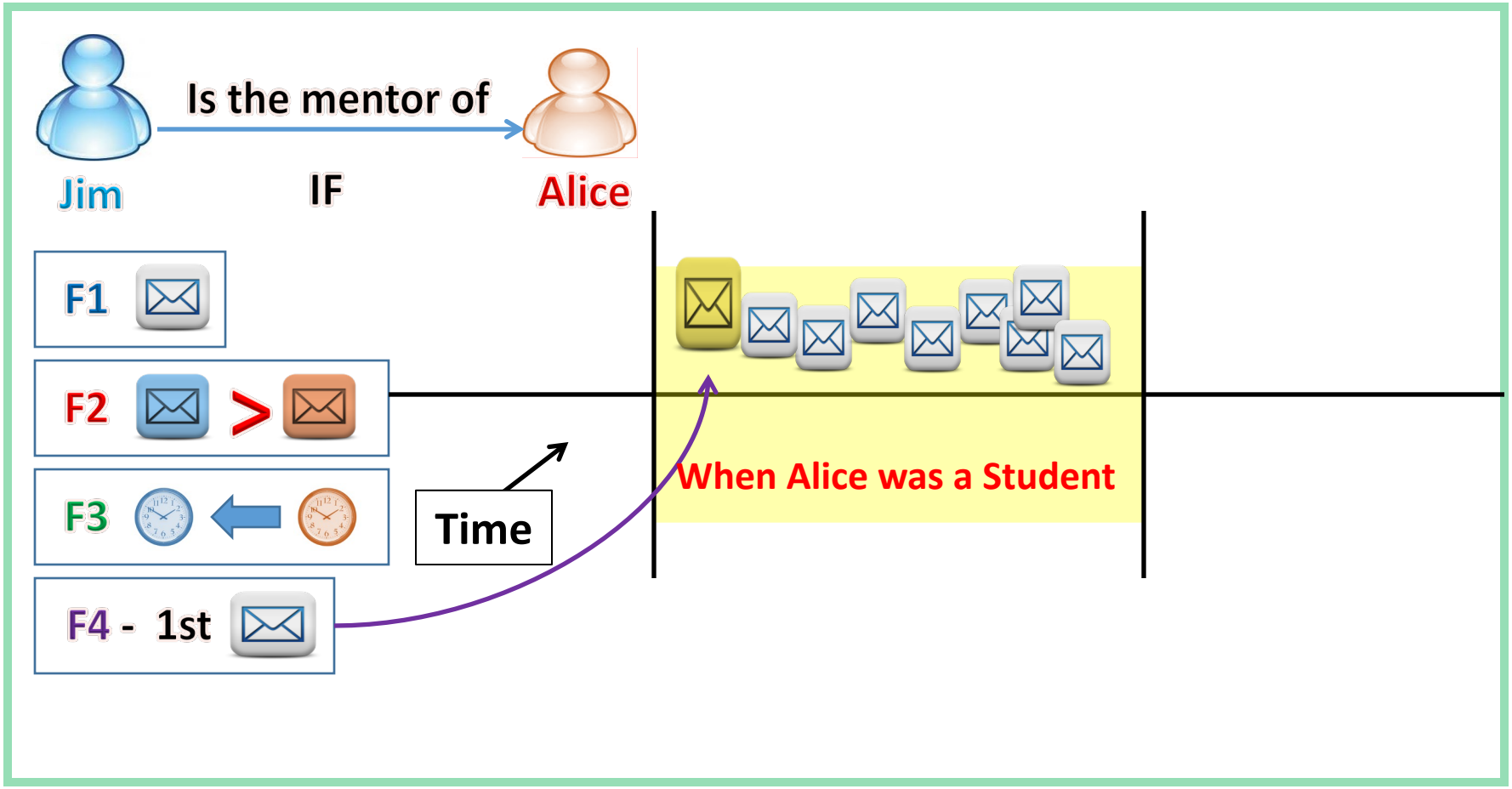
YODA

F3: project age



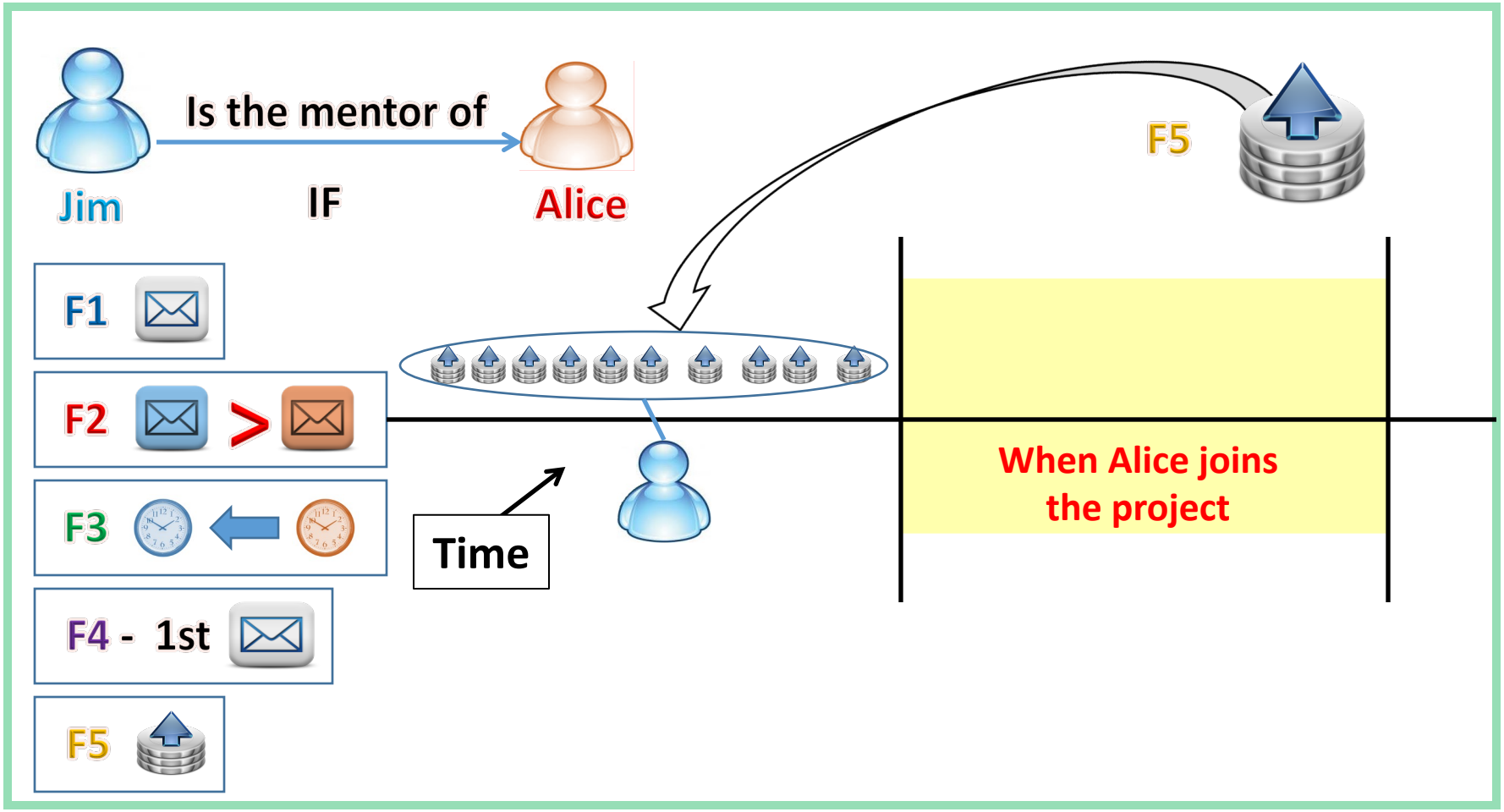
YODA

F4: newcomer early emails



YODA

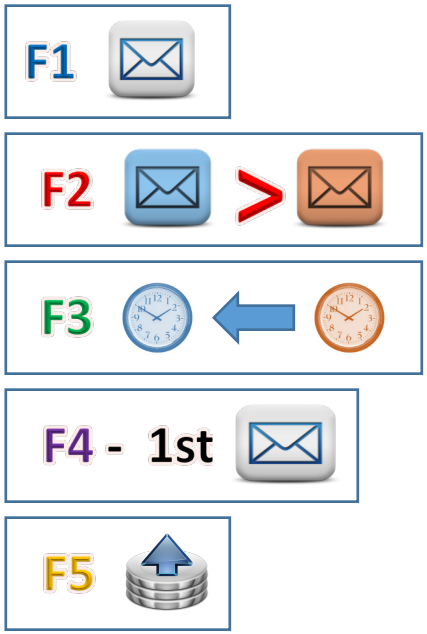
F5: Commits



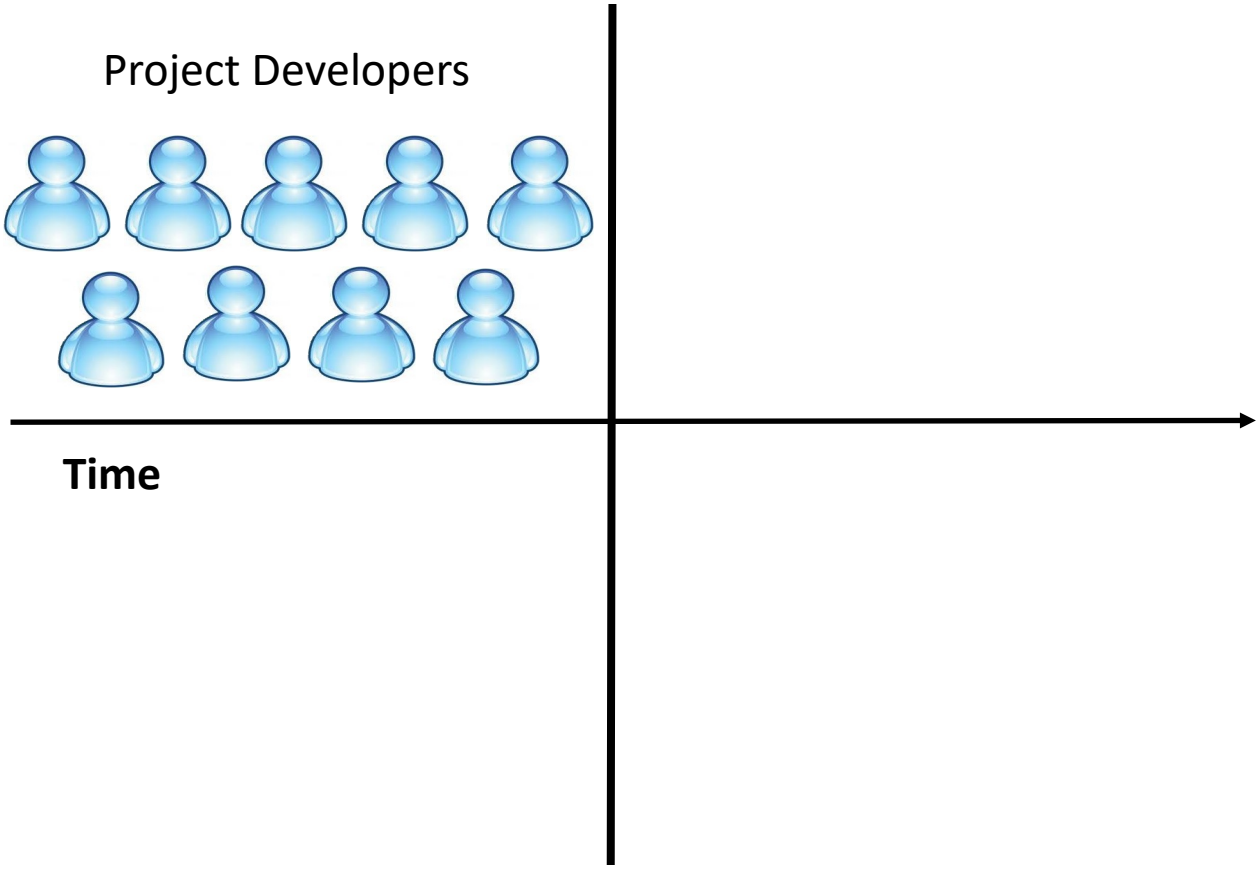
Identify Past Successful Mentors

Score Computed Aggregating the Factors in a Weighted Sum

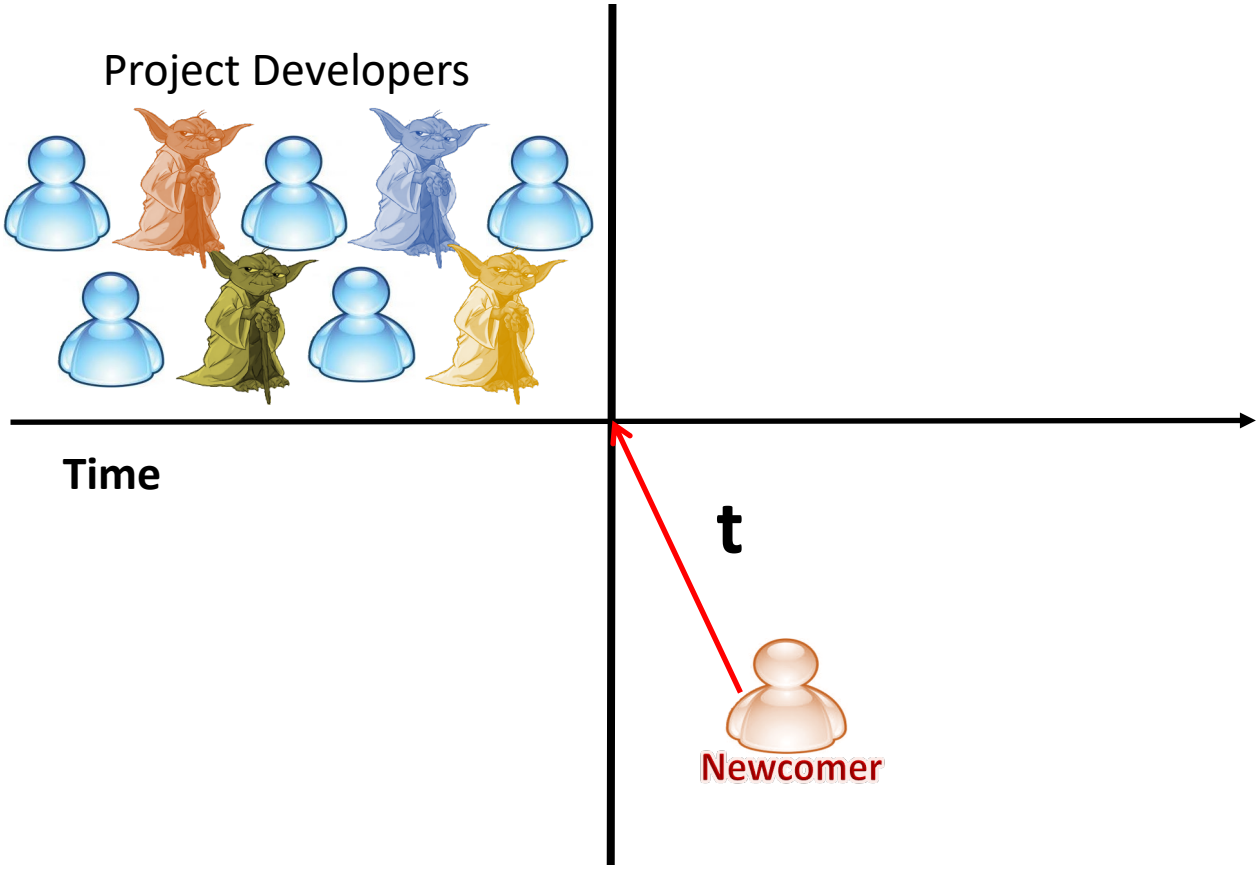
$$\sum_{i=1}^{5} w_i f_i$$



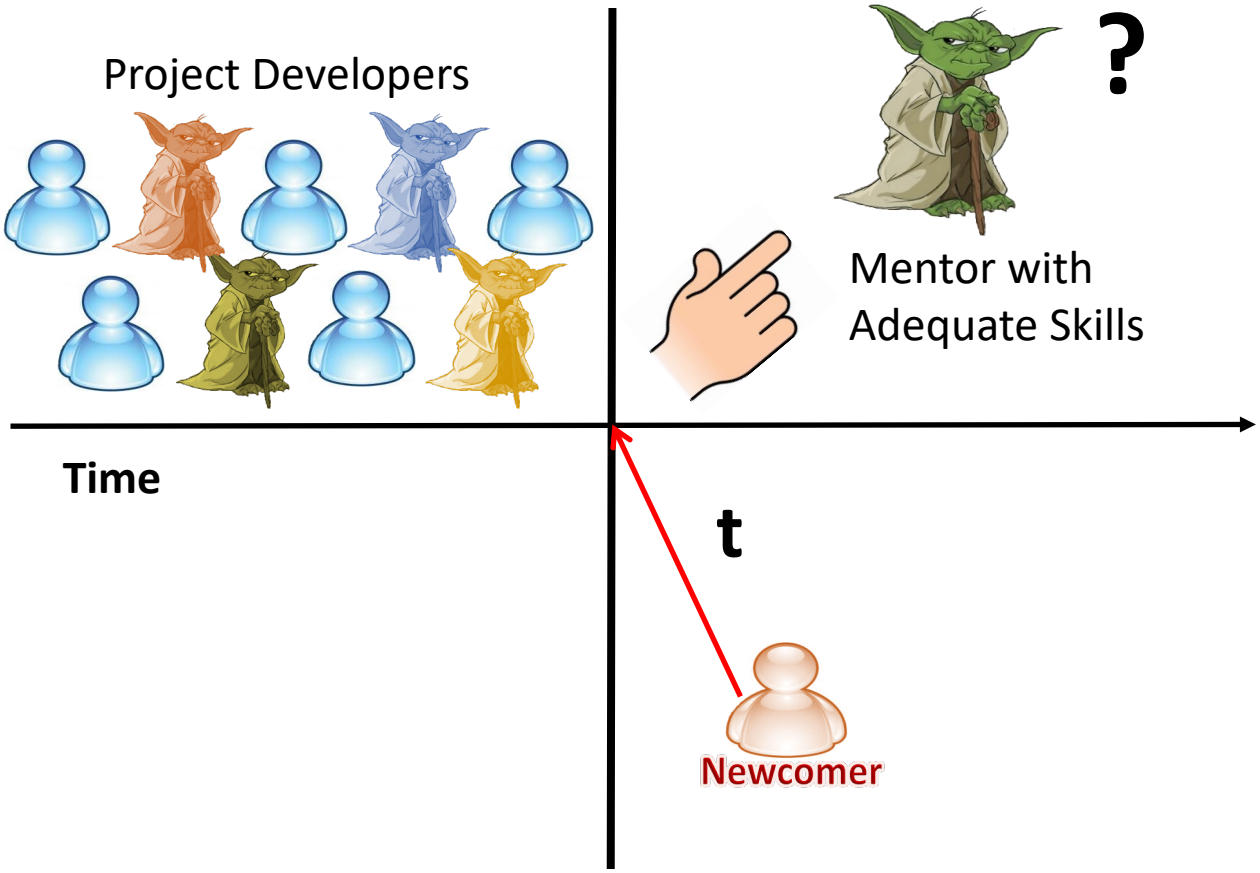
Recommending Mentors



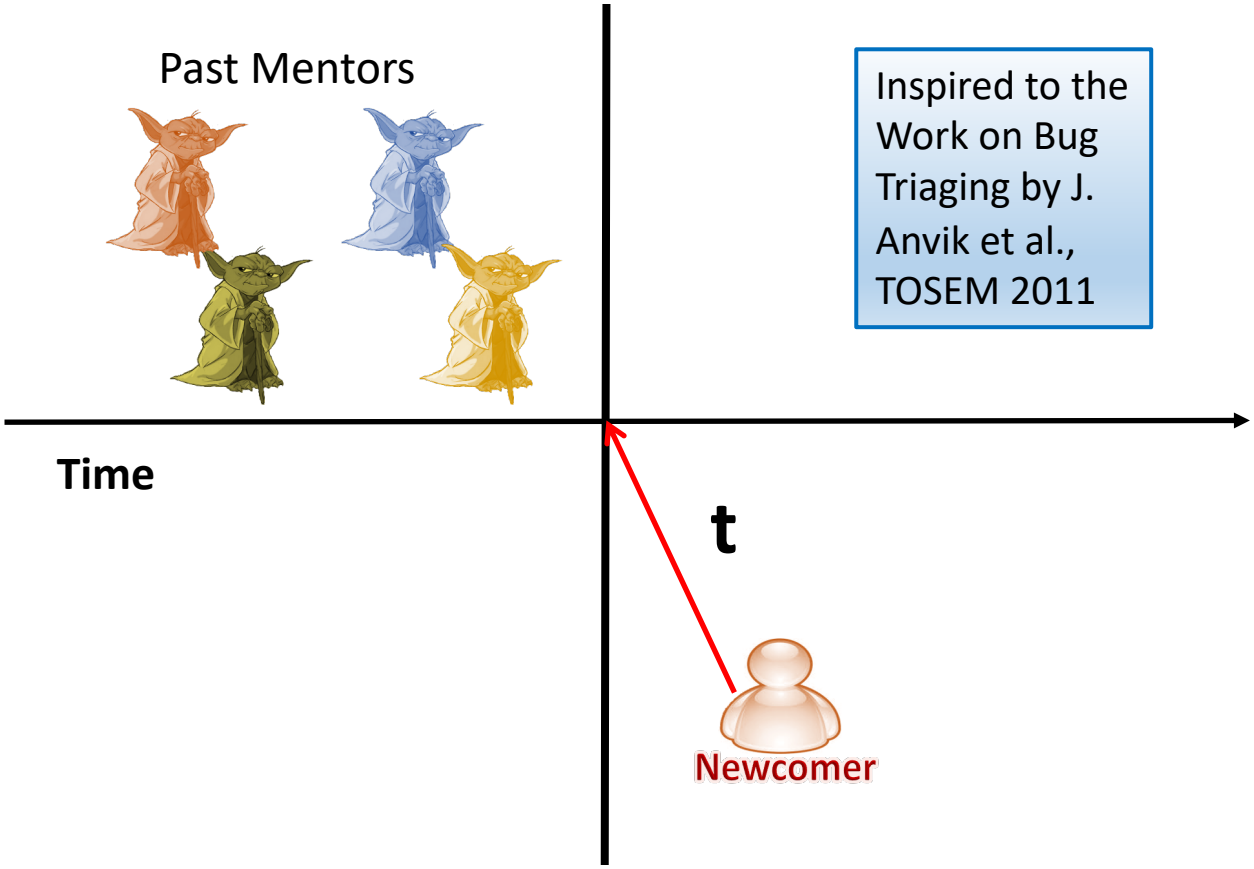
Recommending Mentors



Recommending Mentors

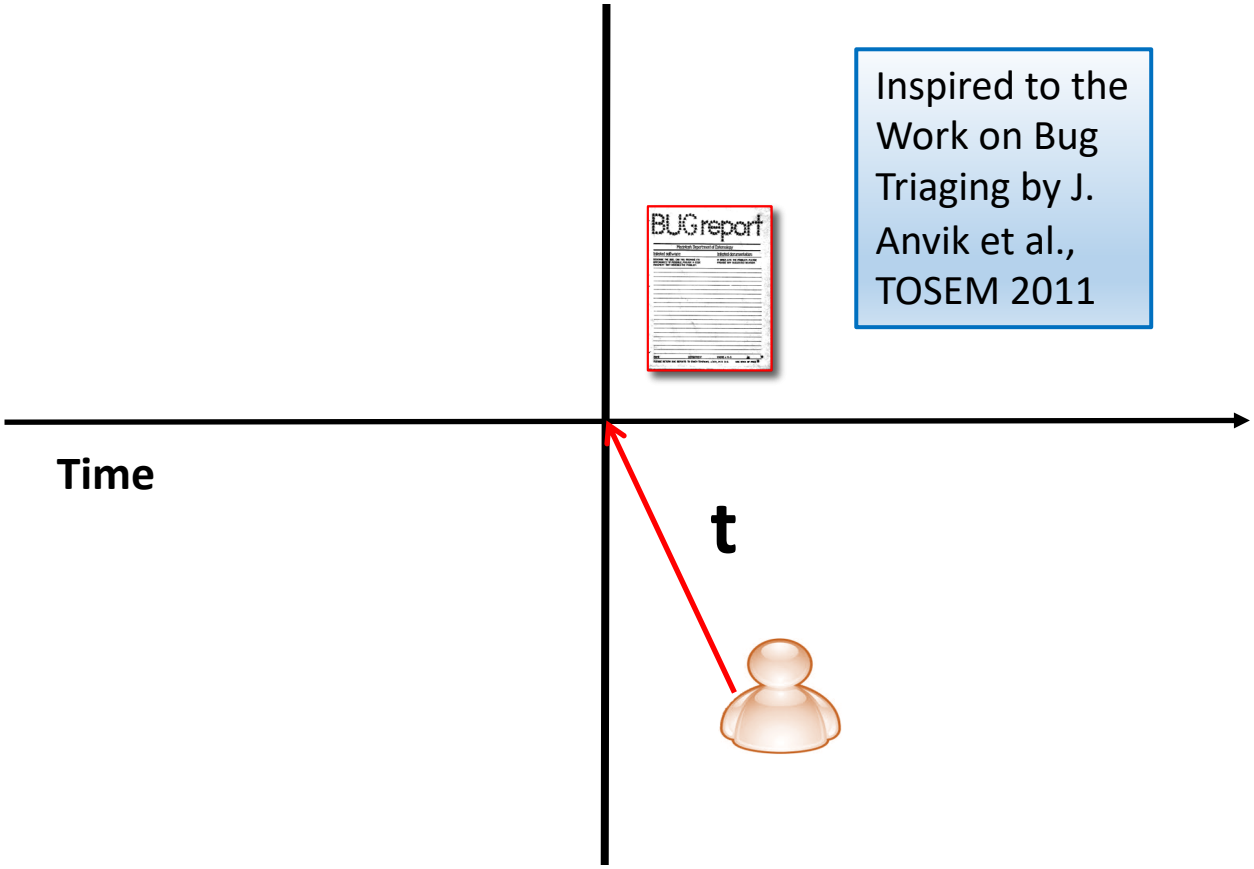


Recommending Mentors



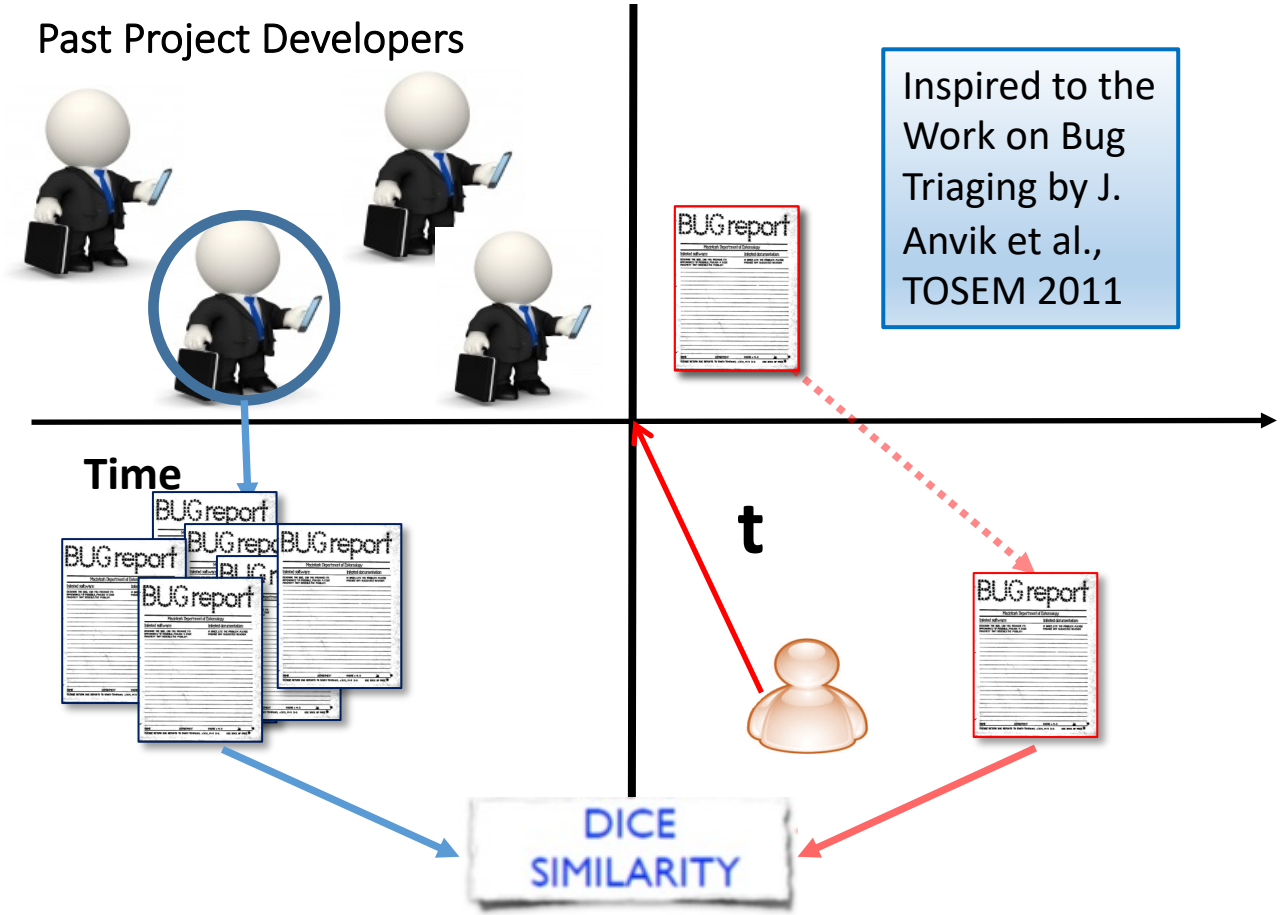
Inspired to the Work on Bug Triaging by J. Anvik et al., TOSEM 2011

Recommending Mentors



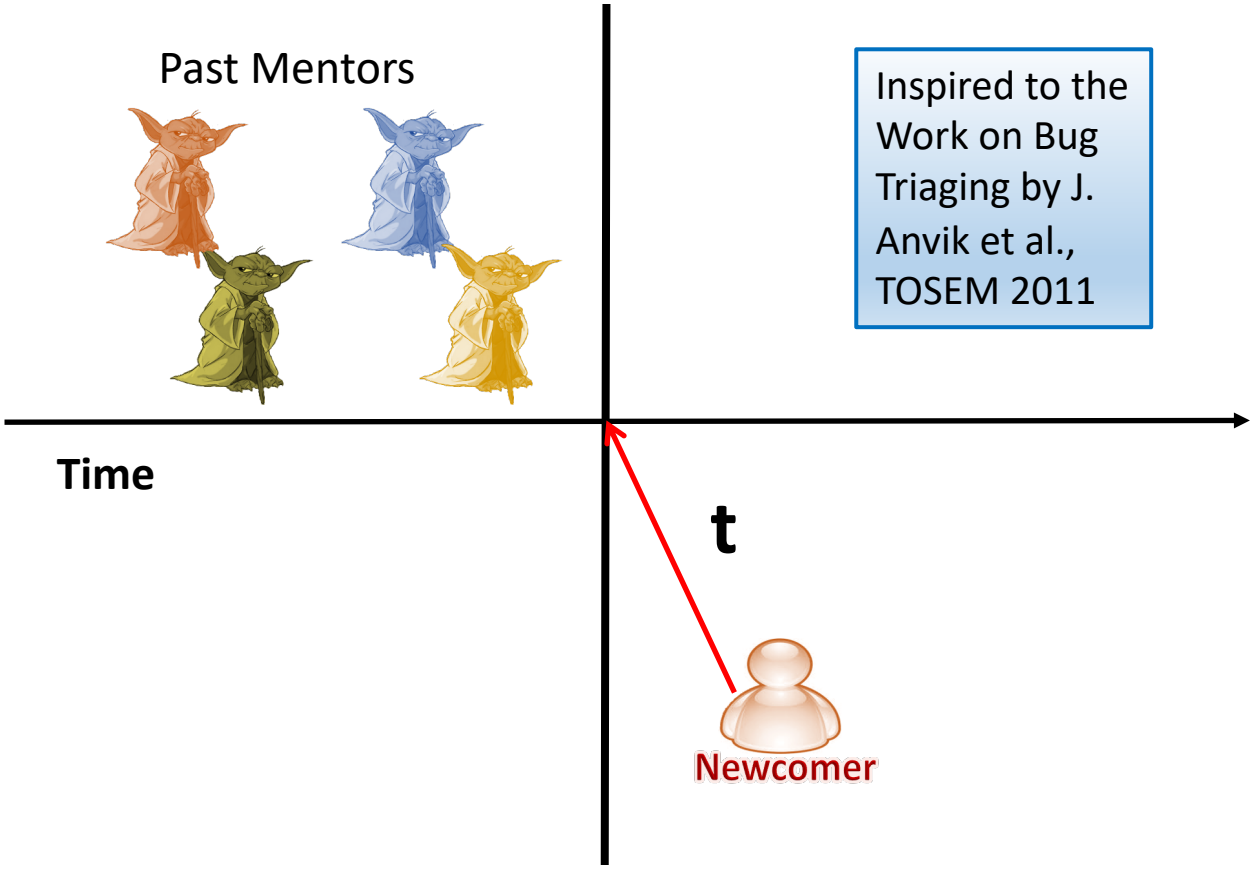
Inspired to the Work on Bug Triaging by J. Anvik et al., TOSEM 2011

Recommending Mentors



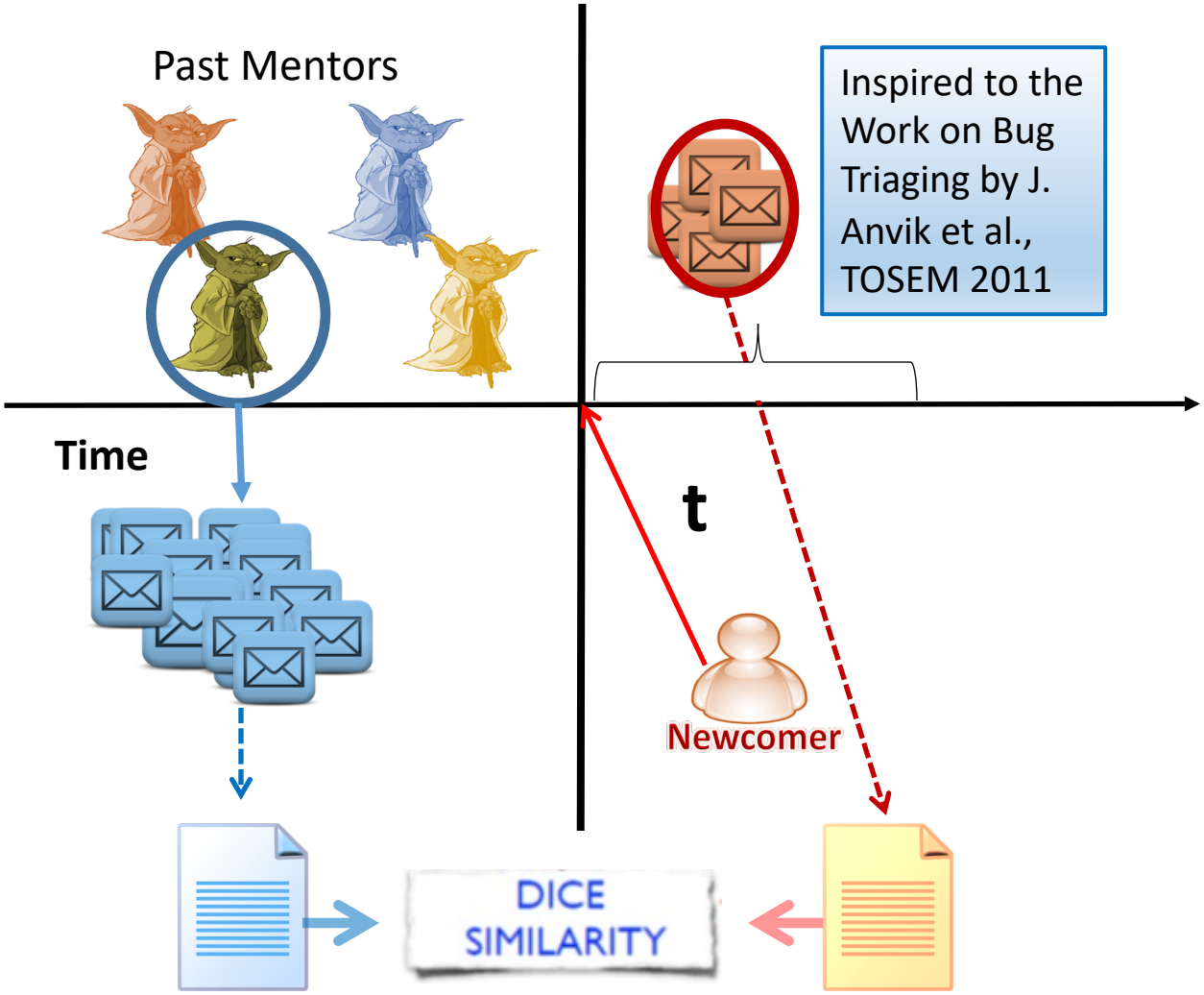
Inspired to the Work on Bug Triaging by J. Anvik et al., TOSEM 2011

Recommending Mentors

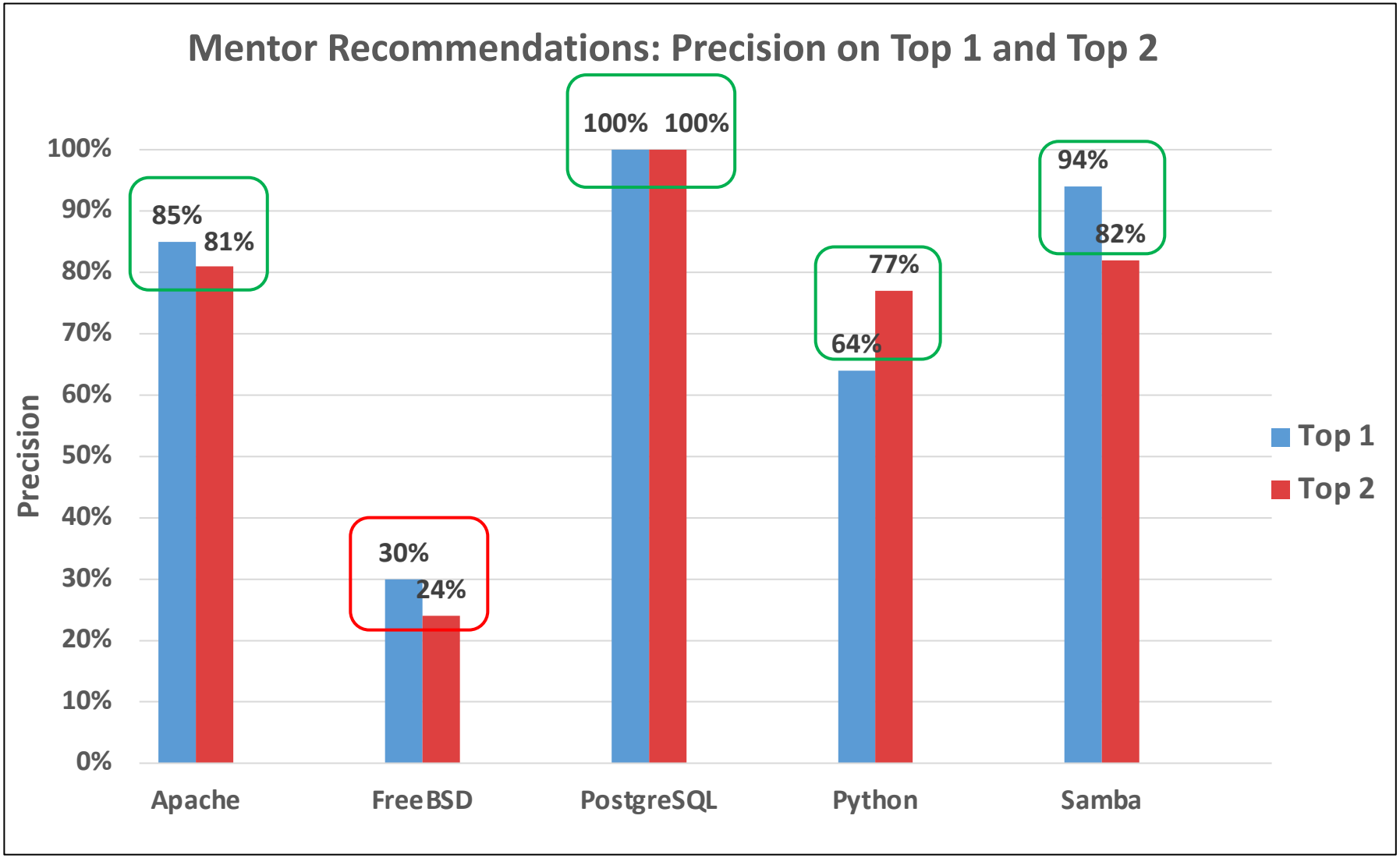


Inspired to the Work on Bug Triaging by J. Anvik et al., TOSEM 2011

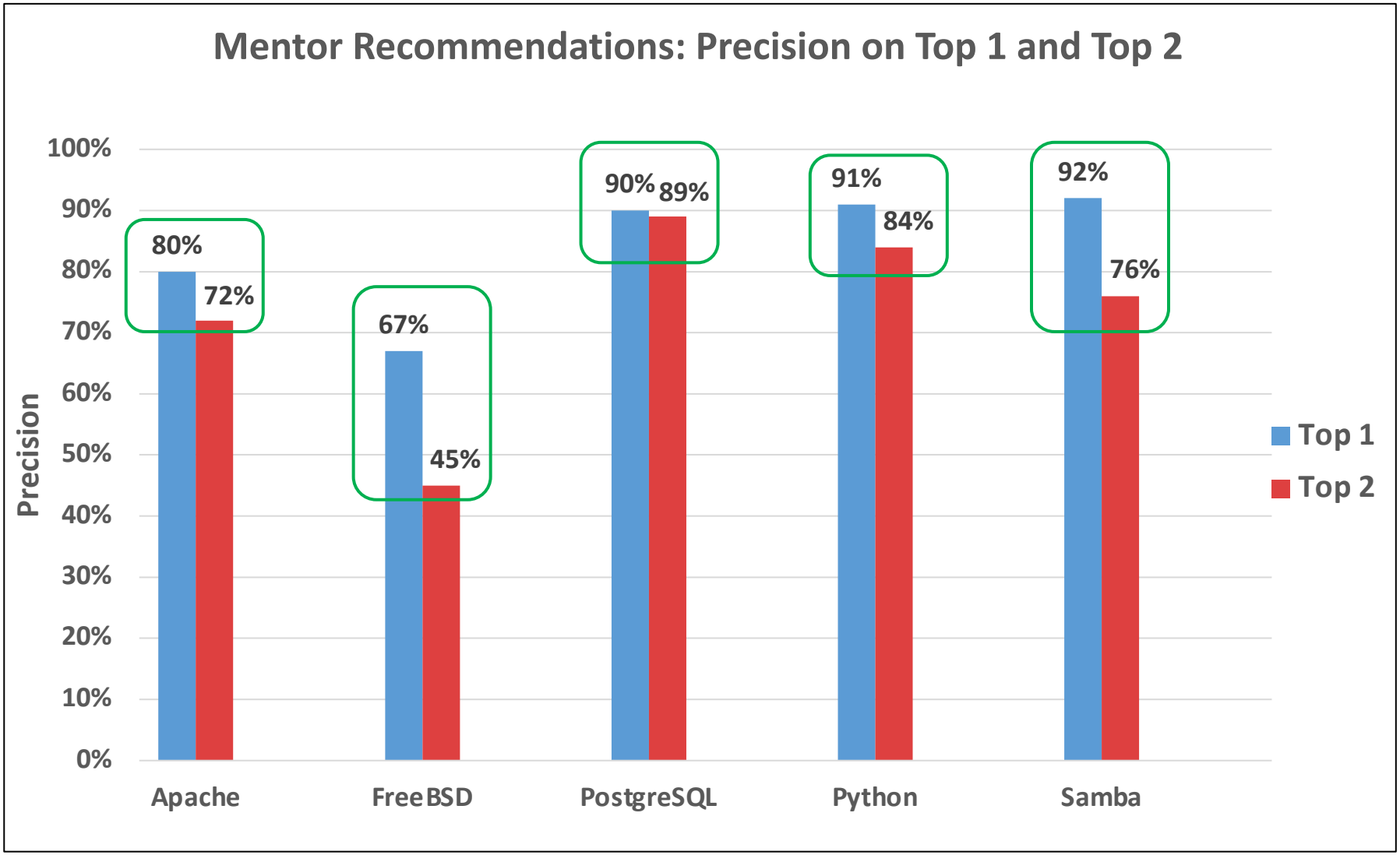
Recommending Mentors



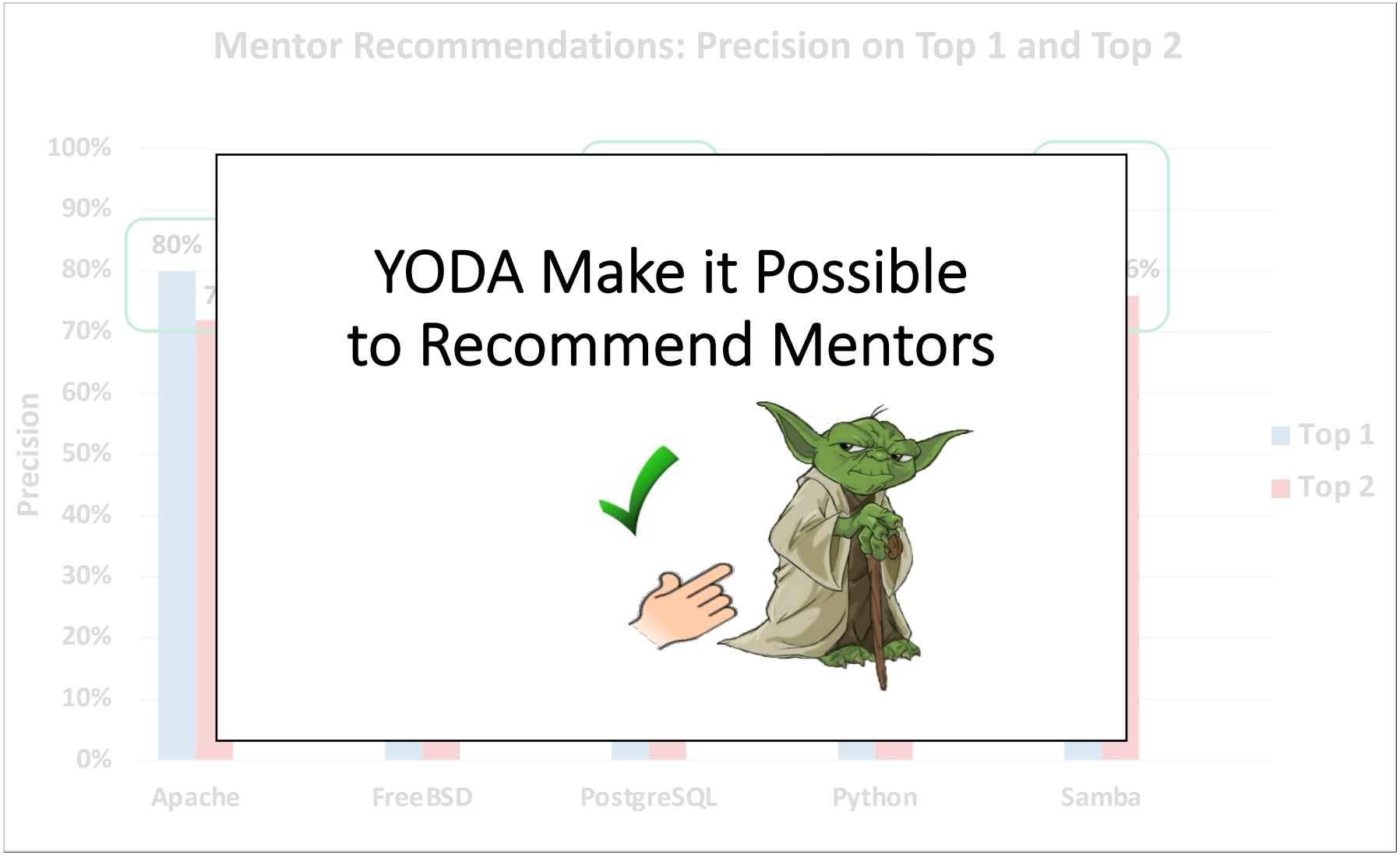
Is it Possible to Recommend Mentors To Project Newcomers?



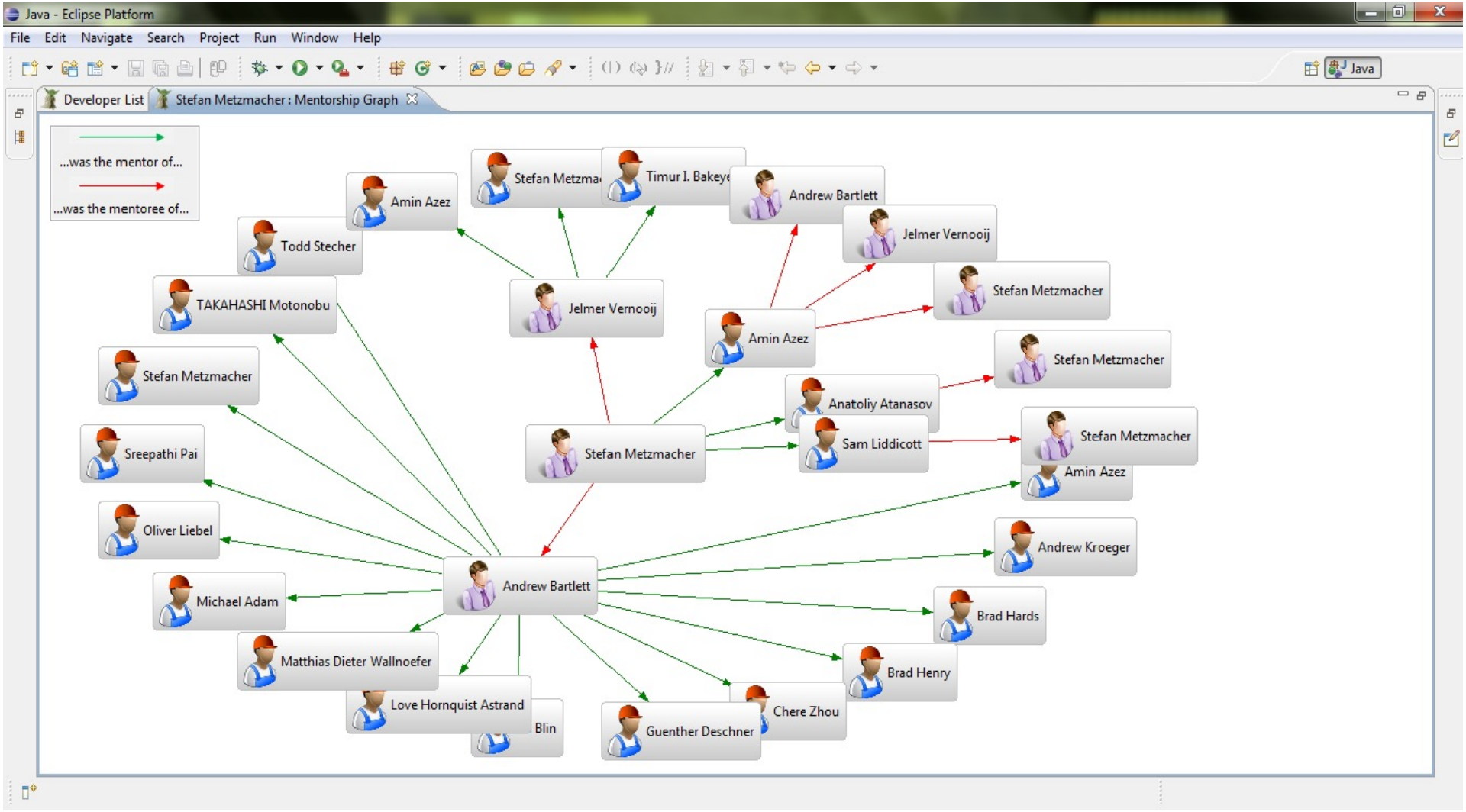
Results When are Used Both Mails and Issues



It is Possible to Recommend Mentors To Project Newcomers?



YODA Tool



http://www.ing.unisannio.it/spanichella/pages/projects.html

YODA Tool

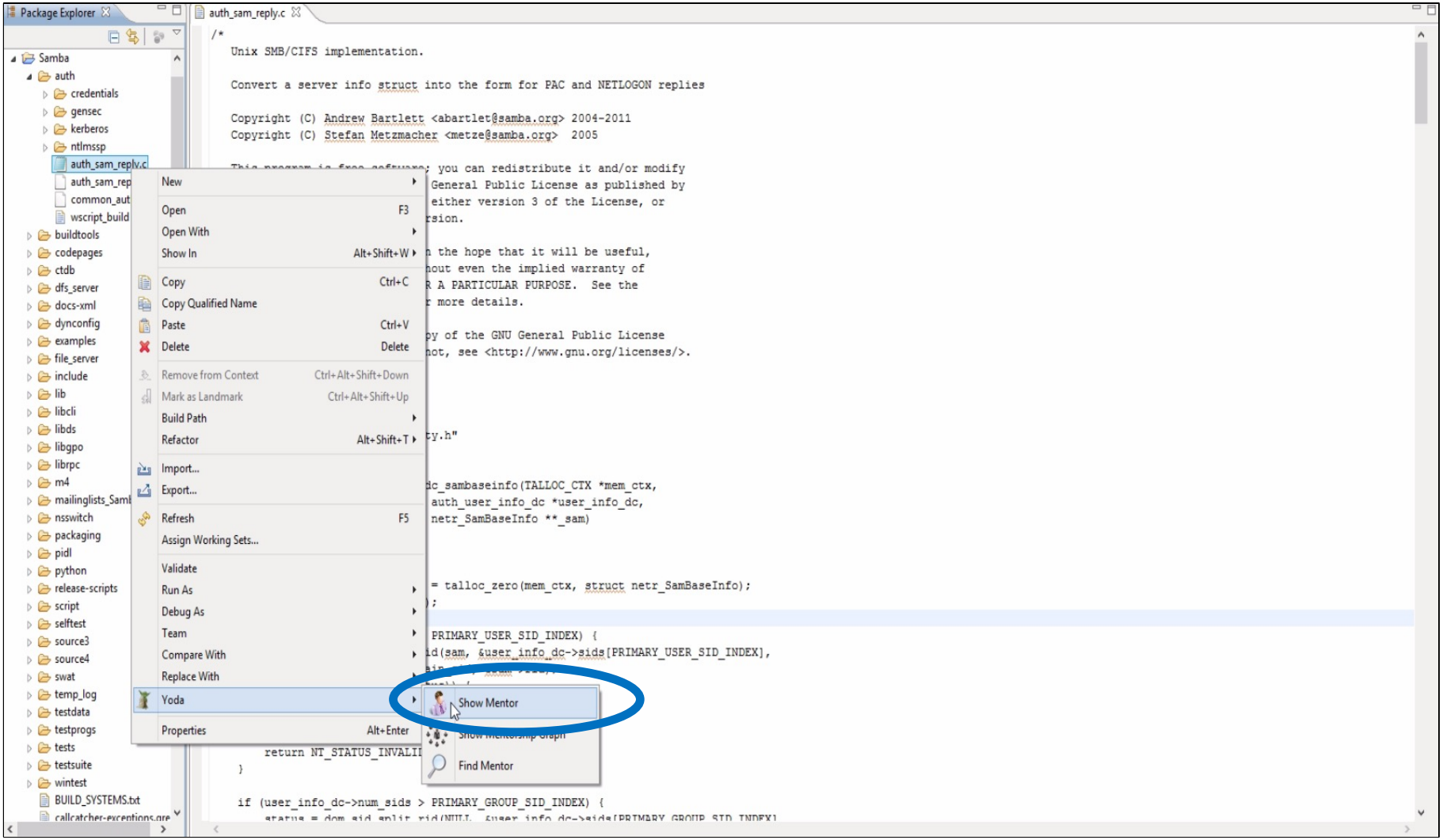
Package Explorer

- Samba
  - auth
    - credentials
    - gensec
    - kerberos
    - auth_sam_reply.c
    - auth_sam_reply.h
    - common_auth.h
    - wscript_build
  - buildtools
  - codepages
  - ctdb
  - dfs_server
  - docs-xml
  - dynconfig
  - examples
  - file_server
  - include
  - lib
  - libcli
  - libds
  - libgpo
  - librpc
  - m4
  - mailinglists_Samba
  - nsswitch
  - packaging
  - pidl
  - python
  - release-scripts
  - script
  - selftest
  - source3
  - source4
  - swat
  - temp_log
  - testdata
  - testprogs
  - tests
  - testsuite
  - wintest
  - BUILD_SYSTEMS.txt
  - callcatcher-exceptions.gre

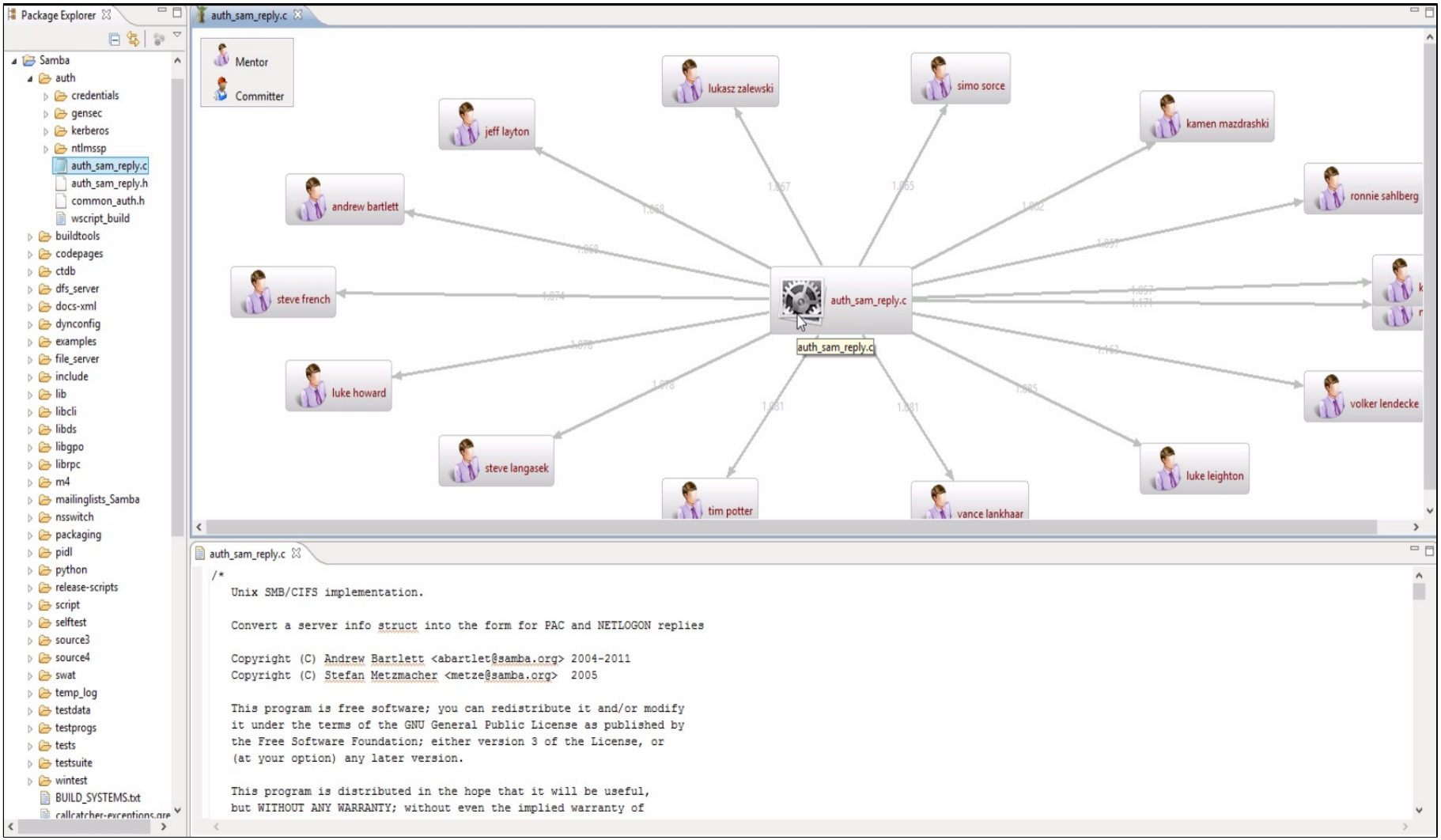
auth_sam_reply.c

```c
/*
   Unix SMB/CIFS implementation.

   Convert a server info struct into the form for PAC and NETLOGON replies

   Copyright (C) Andrew Bartlett <abartlet@samba.org> 2004-2011
   Copyright (C) Stefan Metzmacher <metze@samba.org>  2005

   This program is free software; you can redistribute it and/or modify
   it under the terms of the GNU General Public License as published by
   the Free Software Foundation; either version 3 of the License, or
   (at your option) any later version.

   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   GNU General Public License for more details.

   You should have received a copy of the GNU General Public License
   along with this program.  If not, see <http://www.gnu.org/licenses/>.
*/

#include "includes.h"
#include "librpc/gen_ndr/auth.h"
#include "libcli/security/security.h"
#include "auth/auth_sam_reply.h"

NTSTATUS auth_convert_user_info_dc_sambaseinfo(TALLOC_CTX *mem_ctx,
                        struct auth_user_info_dc *user_info_dc,
                        struct netr_SamBaseInfo **_sam)
{
    NTSTATUS status;
    struct auth_user_info *info;
    struct netr_SamBaseInfo *sam = talloc_zero(mem_ctx, struct netr_SamBaseInfo);
    NT_STATUS_HAVE_NO_MEMORY(sam);

    if (user_info_dc->num_sids > PRIMARY_USER_SID_INDEX) {
        status = dom_sid_split_rid(sam, &user_info_dc->sids[PRIMARY_USER_SID_INDEX],
                    &sam->domain_sid, &sam->rid);
        if (!NT_STATUS_IS_OK(status)) {
            return status;
        }
    } else {
        return NT_STATUS_INVALID_PARAMETER;
    }

    if (user_info_dc->num_sids > PRIMARY_GROUP_SID_INDEX) {
        status = dom_sid_split_rid(NULL, &user_info_dc->sids[PRIMARY_GROUP_SID_INDEX]
```

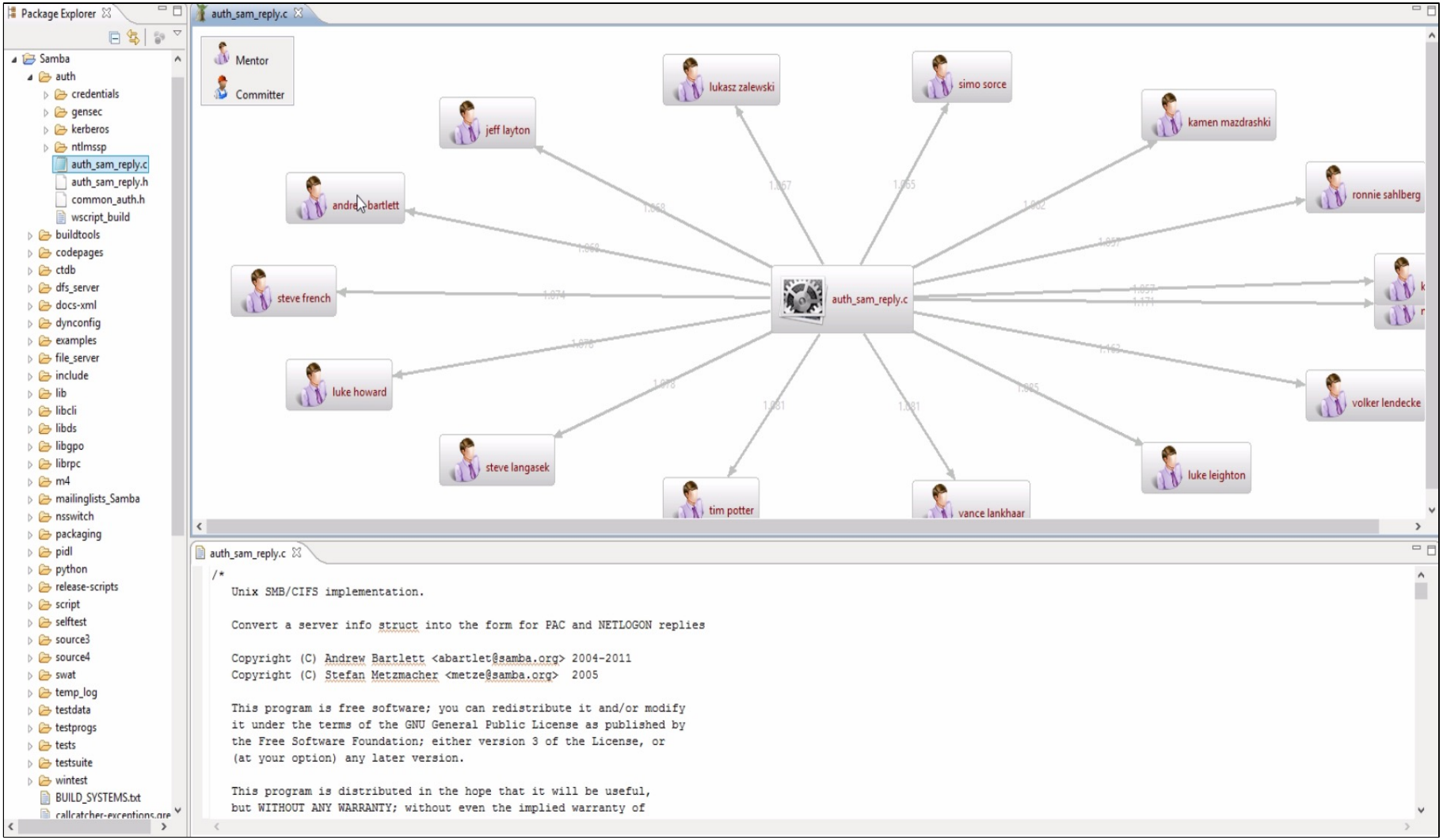
YODA Tool


Package Explorer
auth_sam_reply.c
Samba
auth
credentials
gensec
kerberos
ntlmssp
auth_sam_reply.c
auth_sam_rep
common_aut
wscript_build
buildtools
codepages
ctdb
dfs_server
docs-xml
dynconfig
examples
file_server
include
lib
libcli
libds
libgpo
librpc
m4
mailinglists_Saml
nsswitch
packaging
pidl
python
release-scripts
script
selftest
source3
source4
swat
temp_log
testdata
testprogs
tests
testsuite
wintest
BUILD_SYSTEMS.txt
New
Open F3
Open With
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Mark as Landmark Ctrl+Alt+Shift+Up
Build Path
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Assign Working Sets...
Validate
Run As
Debug As
Team
Compare With
Replace With
Yoda
Properties Alt+Enter
Show Mentor
Find Mentor
Unix SMB/CIFS implementation.
Convert a server info struct into the form for PAC and NETLOGON replies
Copyright (C) Andrew Bartlett <abartlet@samba.org> 2004-2011
Copyright (C) Stefan Metzmacher <metze@samba.org> 2005

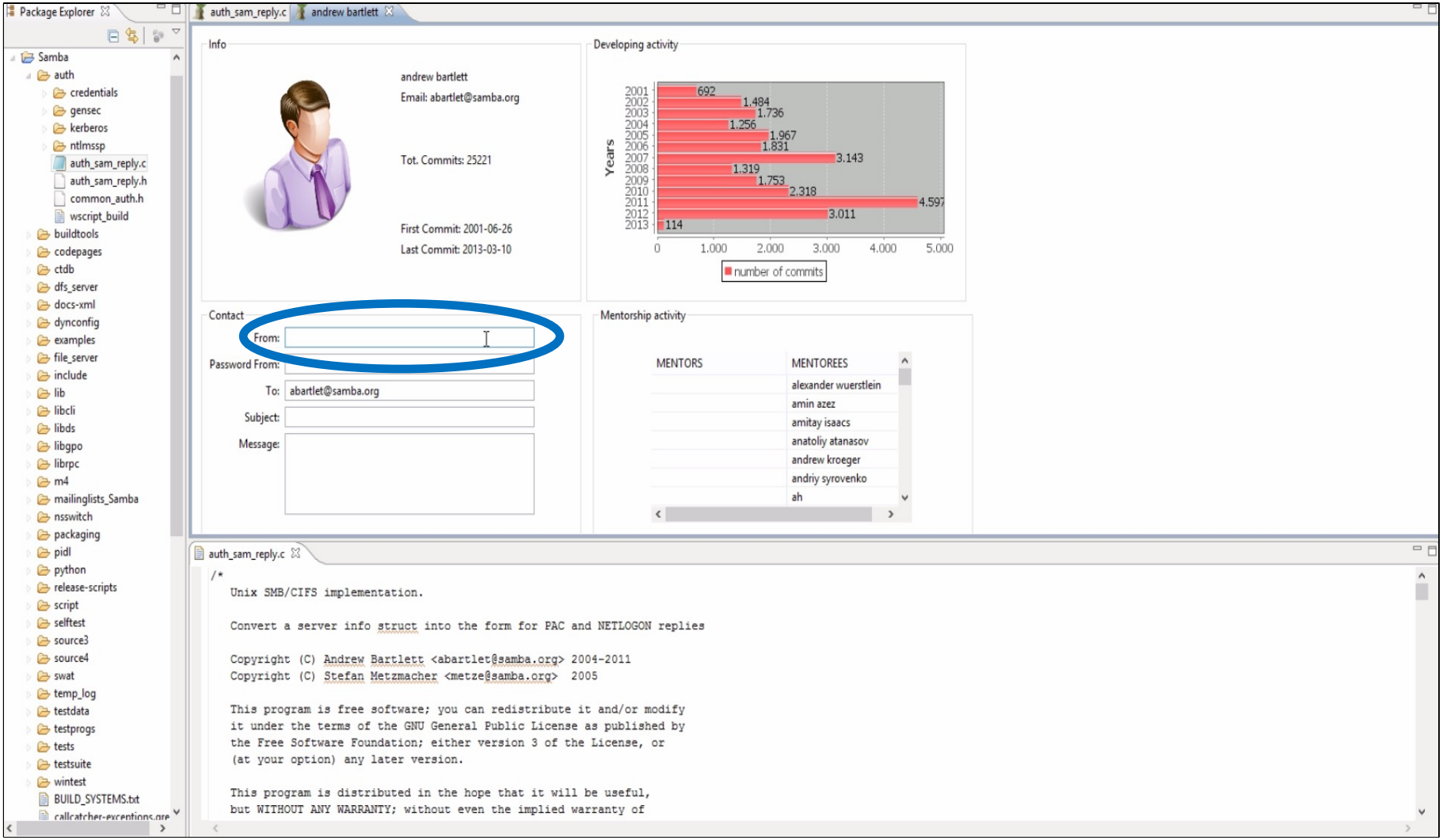
YODA Tool


Package Explorer
Samba
auth
credentials
gensec
kerberos
ntlmssp
auth_sam_reply.c
auth_sam_reply.h
common_auth.h
wscript_build
buildtools
codepages
ctdb
dfs_server
docs-xml
dynconfig
examples
file_server
include
lib
libcli
libds
libgpo
librpc
m4
mailinglists_Samba
nsswitch
packaging
pidl
python
release-scripts
script
selftest
source3
source4
swat
temp_log
testdata
testprogs
tests
testsuite
wintest
BUILD_SYSTEMS.txt
auth_sam_reply.c
Mentor
Committer
lukasz zalewski
simo sorce
jeff layton
kamen mazdrashki
andrew bartlett
ronnie sahlberg
steve french
auth_sam_reply.c
luke howard
volker lendecke
steve langasek
luke leighton
tim potter
vance lankhaar
auth_sam_reply.c
/*
Unix SMB/CIFS implementation.
Convert a server info struct into the form for PAC and NETLOGON replies
Copyright (C) Andrew Bartlett <abartlet@samba.org> 2004-2011
Copyright (C) Stefan Metzmacher <metze@samba.org> 2005
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 3 of the License, or
(at your option) any later version.
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of

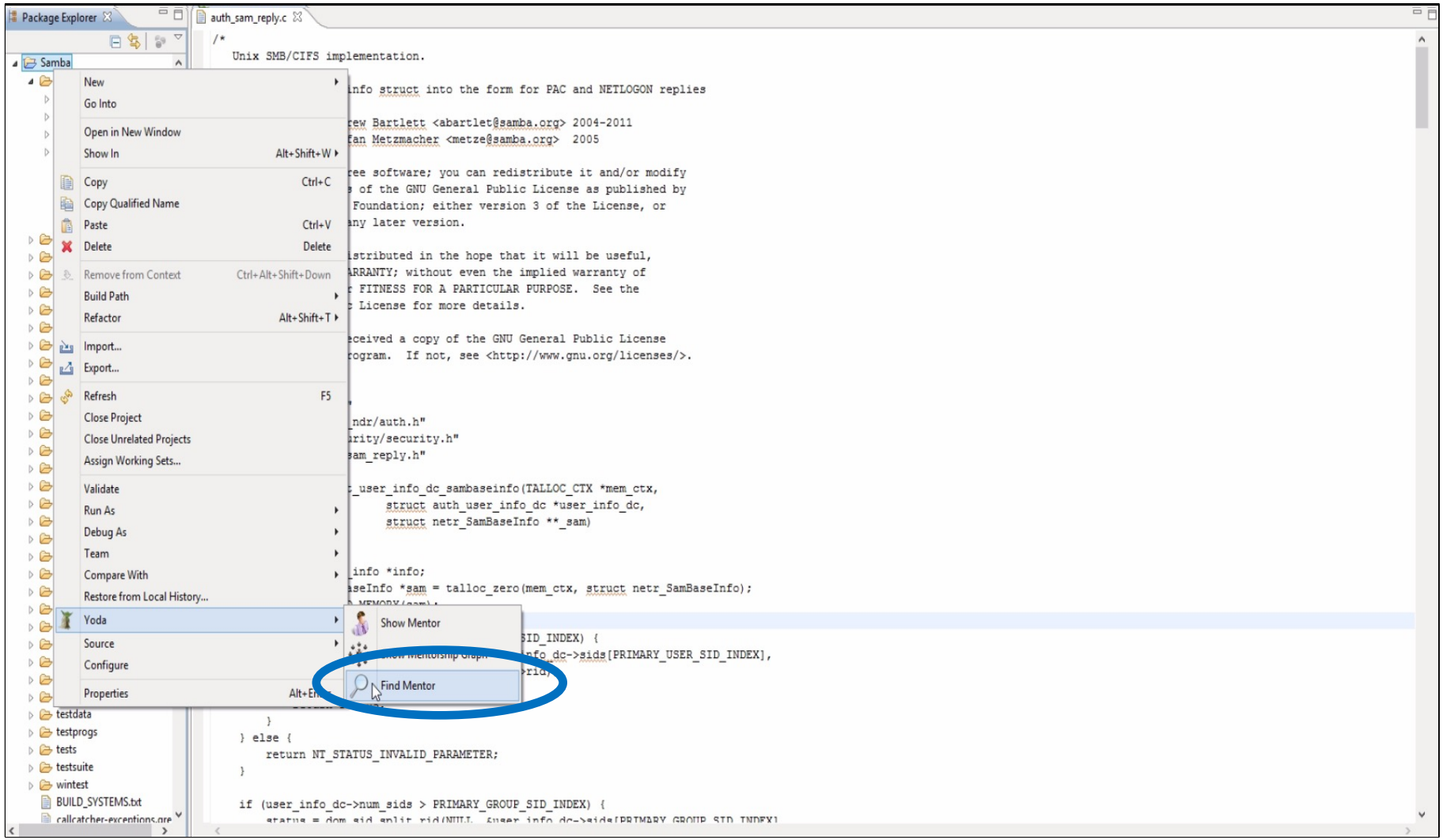
YODA Tool


Package Explorer
Samba
auth
credentials
gensec
kerberos
ntlmssp
auth_sam_reply.c
auth_sam_reply.h
common_auth.h
wscript_build
buildtools
codepages
ctdb
dfs_server
docs-xml
dynconfig
examples
file_server
include
lib
libcli
libds
libgpo
librpc
m4
mailinglists_Samba
nsswitch
packaging
pidl
python
release-scripts
script
selftest
source3
source4
swat
temp_log
testdata
testprogs
tests
testsuite
wintest
BUILD_SYSTEMS.txt
auth_sam_reply.c
Mentor
Committer
lukasz zalewski
simo sorce
jeff layton
kamen mazdrashki
andrew bartlett
ronnie sahlberg
steve french
auth_sam_reply.c
luke howard
volker lendecke
steve langasek
luke leighton
tim potter
vance lankhaar
auth_sam_reply.c
/*
Unix SMB/CIFS implementation.
Convert a server info struct into the form for PAC and NETLOGON replies
Copyright (C) Andrew Bartlett <abartlet@samba.org> 2004-2011
Copyright (C) Stefan Metzmacher <metze@samba.org> 2005
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 3 of the License, or
(at your option) any later version.
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of

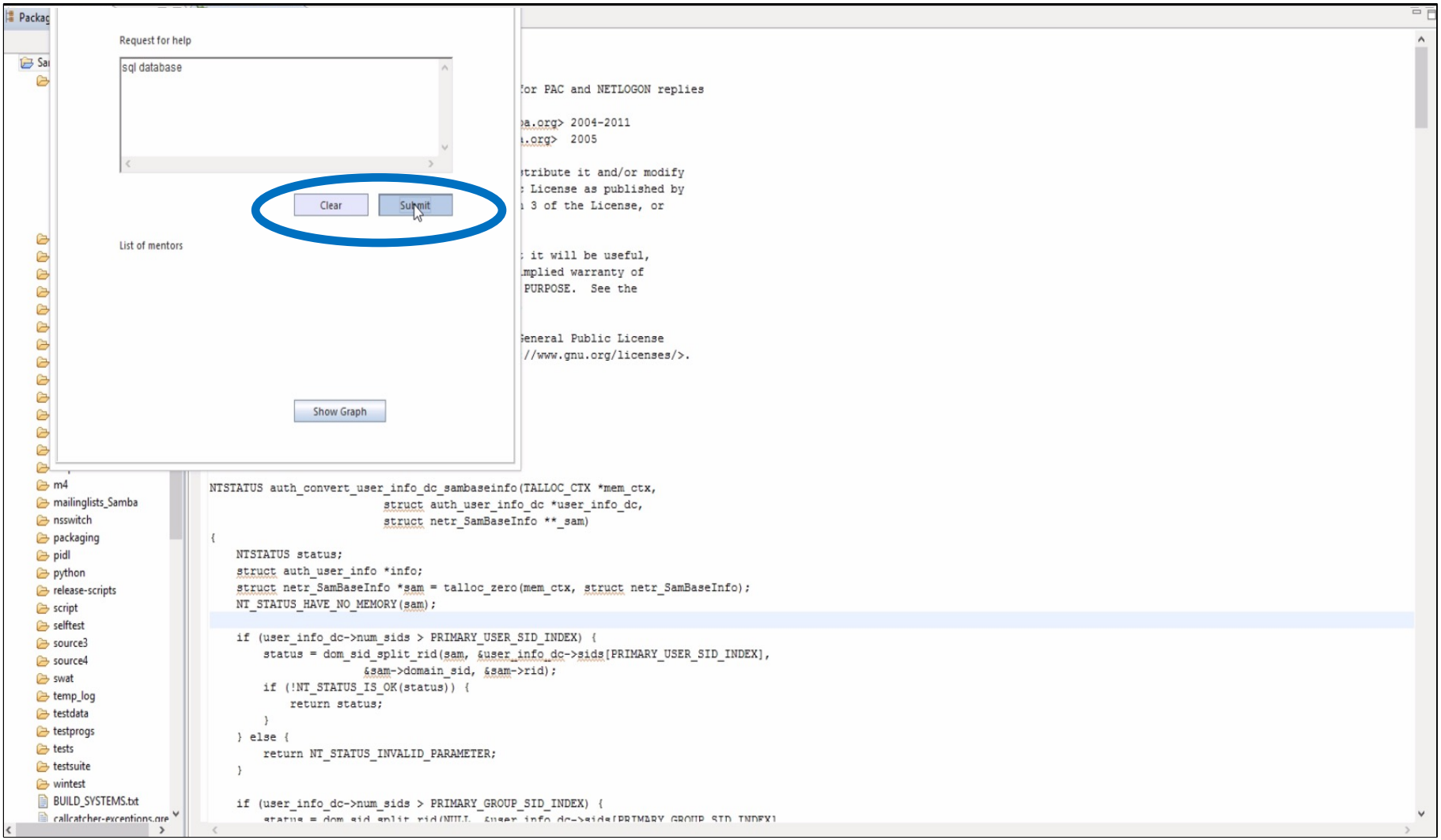
YODA Tool


Package Explorer
Samba
auth
credentials
gensec
kerberos
ntlmssp
auth_sam_reply.c
auth_sam_reply.h
common_auth.h
wscript_build
buildtools
codepages
ctdb
dfs_server
docs-xml
dynconfig
examples
file_server
include
lib
libcli
libds
libgpo
librpc
m4
mailinglists_Samba
nsswitch
packaging
pidl
python
release-scripts
script
selftest
source3
source4
swat
temp_log
testdata
testprogs
tests
testsuite
wintest
BUILD_SYSTEMS.txt
auth_sam_reply.c
andrew bartlett
Info
andrew bartlett
Email: abartlet@samba.org
Tot. Commits: 25221
First Commit: 2001-06-26
Last Commit: 2013-03-10
Developing activity
Years
2001 692
2002 1.484
2003 1.736
2004 1.256
2005 1.967
2006 1.831
2007 3.143
2008 1.319
2009 1.753
2010 2.318
2011 4.597
2012 3.011
2013 114
0 1.000 2.000 3.000 4.000 5.000
number of commits
Contact
From:
Password From:
To: abartlet@samba.org
Subject:
Message:
Mentorship activity
MENTORS
MENTOREES
alexander wuerstlein
amin azez
amitay isaacs
anatoliy atanasov
andrew kroeger
andriy syrovenko
ah
auth_sam_reply.c
/*
Unix SMB/CIFS implementation.
Convert a server info struct into the form for PAC and NETLOGON replies
Copyright (C) Andrew Bartlett <abartlet@samba.org> 2004-2011
Copyright (C) Stefan Metzmacher <metze@samba.org> 2005
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 3 of the License, or
(at your option) any later version.
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of

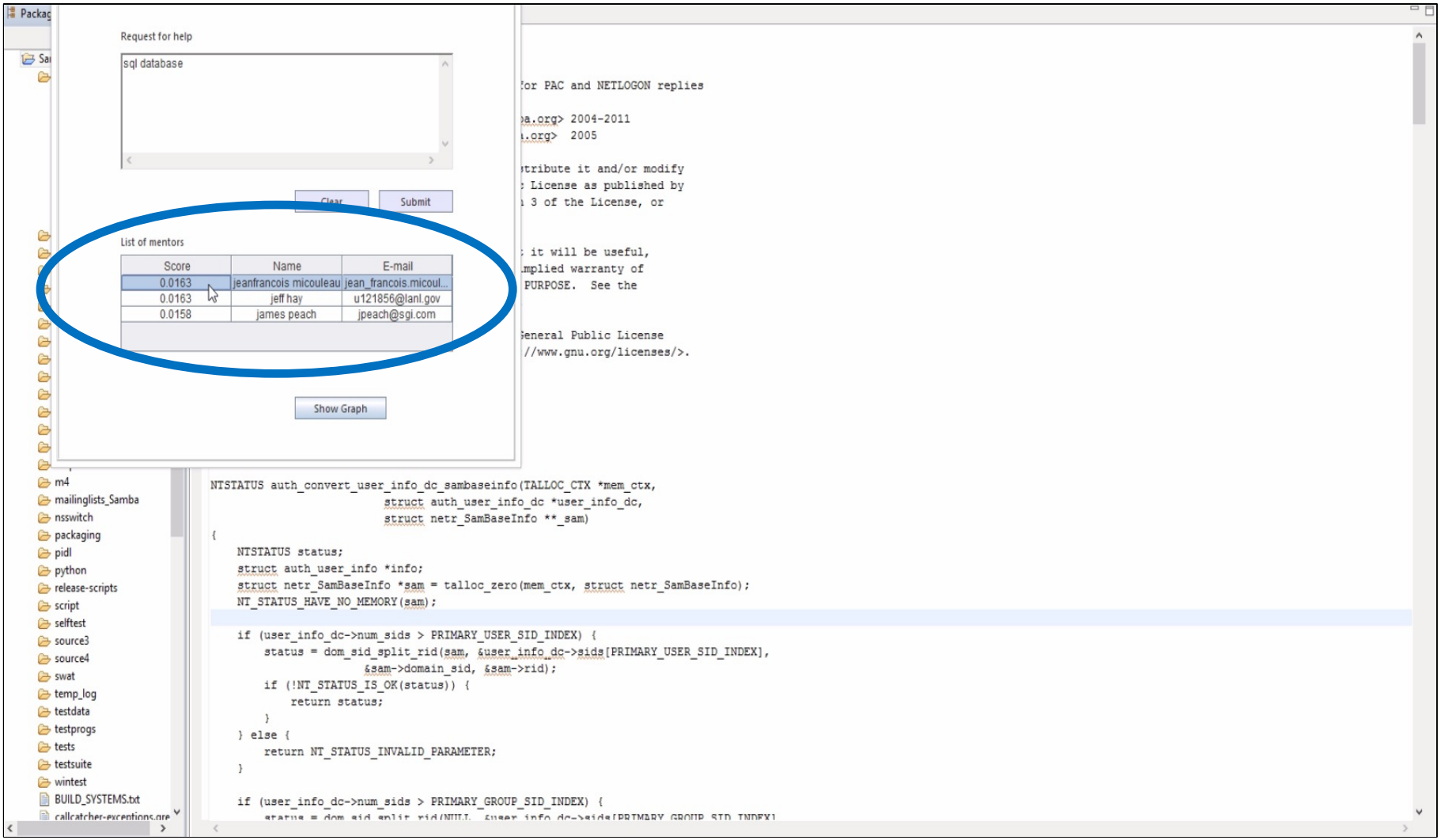
YODA Tool


Package Explorer
auth_sam_reply.c
Samba
New
Go Into
Open in New Window
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Validate
Run As
Debug As
Team
Compare With
Restore from Local History...
Yoda
Source
Configure
Properties
Show Mentor
Find Mentor
testdata
testprogs
tests
testsuite
wintest
BUILD_SYSTEMS.txt

YODA Tool


Request for help
sql database
Clear
Submit
List of mentors
Show Graph
m4
mailinglists_Samba
nsswitch
packaging
pidl
python
release-scripts
script
selftest
source3
source4
swat
temp_log
testdata
testprogs
tests
testsuite
wintest
BUILD_SYSTEMS.txt
NTSTATUS auth_convert_user_info_dc_sambaseinfo(TALLOC_CTX *mem_ctx,
struct auth_user_info_dc *user_info_dc,
struct netr_SamBaseInfo **_sam)
{
NTSTATUS status;
struct auth_user_info *info;
struct netr_SamBaseInfo *sam = talloc_zero(mem_ctx, struct netr_SamBaseInfo);
NT_STATUS_HAVE_NO_MEMORY(sam);
if (user_info_dc->num_sids > PRIMARY_USER_SID_INDEX) {
status = dom_sid_split_rid(sam, &user_info_dc->sids[PRIMARY_USER_SID_INDEX],
&sam->domain_sid, &sam->rid);
if (!NT_STATUS_IS_OK(status)) {
return status;
}
} else {
return NT_STATUS_INVALID_PARAMETER;
}
if (user_info_dc->num_sids > PRIMARY_GROUP_SID_INDEX) {

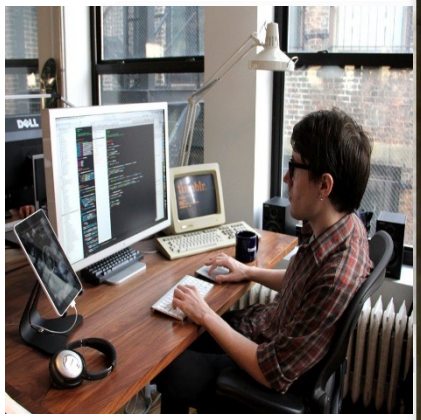
YODA Tool


Request for help
sql database
Clear
Submit
List of mentors
Score
Name
E-mail
0.0163
jeanfrancois micouleau
jean_francois.micoul...
0.0163
jeff hay
u121856@lanl.gov
0.0158
james peach
jpeach@sgi.com
Show Graph
m4
mailinglists_Samba
nsswitch
packaging
pidl
python
release-scripts
script
selftest
source3
source4
swat
temp_log
testdata
testprogs
tests
testsuite
wintest
BUILD_SYSTEMS.txt
NTSTATUS auth_convert_user_info_dc_sambaseinfo(TALLOC_CTX *mem_ctx,
struct auth_user_info_dc *user_info_dc,
struct netr_SamBaseInfo **_sam)
{
NTSTATUS status;
struct auth_user_info *info;
struct netr_SamBaseInfo *sam = talloc_zero(mem_ctx, struct netr_SamBaseInfo);
NT_STATUS_HAVE_NO_MEMORY(sam);
if (user_info_dc->num_sids > PRIMARY_USER_SID_INDEX) {
status = dom_sid_split_rid(sam, &user_info_dc->sids[PRIMARY_USER_SID_INDEX],
&sam->domain_sid, &sam->rid);
if (!NT_STATUS_IS_OK(status)) {
return status;
}
} else {
return NT_STATUS_INVALID_PARAMETER;
}
if (user_info_dc->num_sids > PRIMARY_GROUP_SID_INDEX) {

PART III – B)

Mining Source Code Descriptions from Developer Communications to Improve Newcomers Program Comprehension

Effort in Program Comprehension



Developers spend more time reading than writing code

We argue th ce
of informatic

In such situ

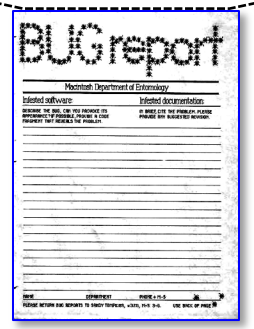
the sour

```
.....................................................
When call the method IndexSplitter.split(File
destDir, String[] segs) from the Lucene cotrib
directory(contrib/misc/src/java/org/apache/luc
ene/index) it creates an index with segments
descriptor file with wrong data. Namely wrong
is the number representing the name of segment
that would be created next in this index.
.....................................................
```

CLASS: `IndexSplitter` **METHOD:** `split`

Newcomer

Can find Source code description



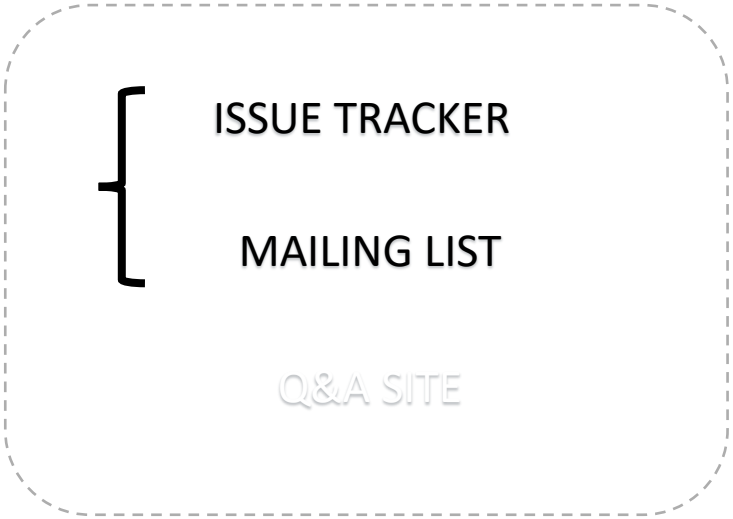
A Five Step-Approach for Mining Method Descriptions

- Step 1: Downloading emails/bugs reports and tracing them onto classes
- Step 2: Extracting paragraphs
- Step 3: Tracing paragraphs onto methods
- Step 4: Heuristic based Filtering
- Step 5: Similarity based Filtering

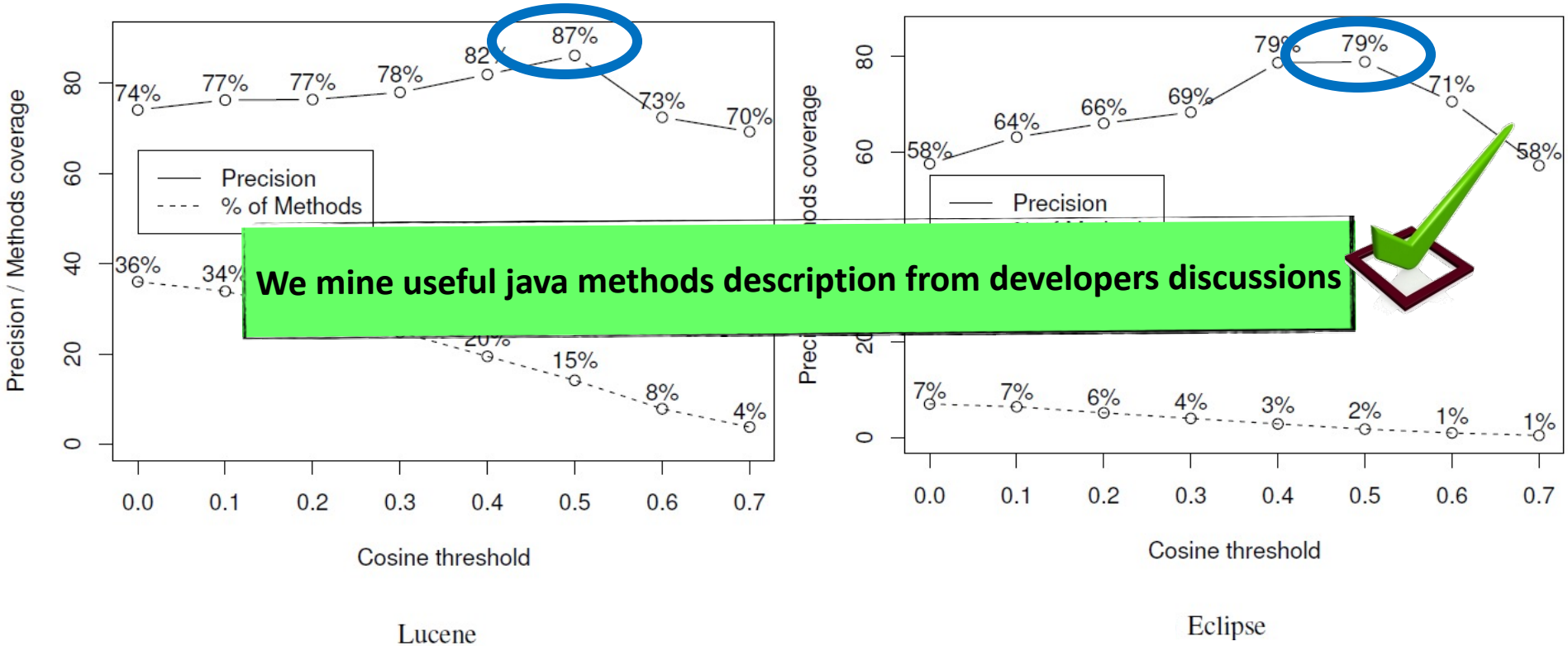
Sebastiano Panichella, Jairo Aponte, Massimiliano Di Penta, Andrian Marcus, Gerardo Canfora: *Mining source code descriptions from developer communications.* International Conference on Program Comprehension (IEEE ICPC 2012)

Supporting Software Development

Help Newcomer Program Comprehension with extraction of summaries of code elements from



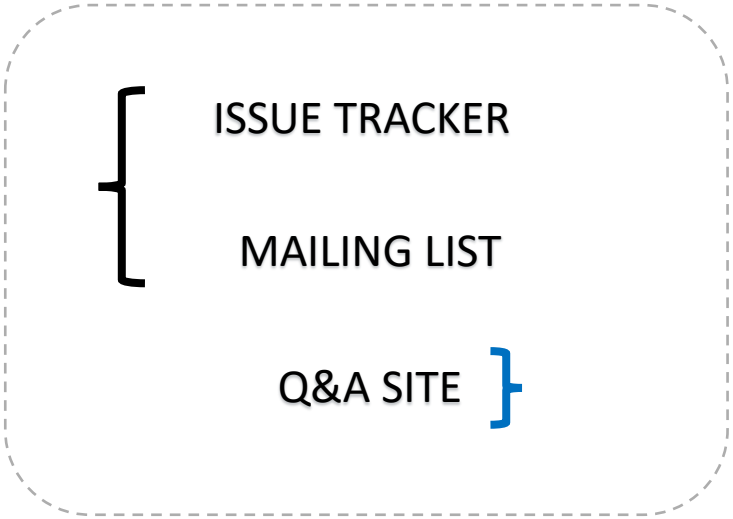
Approach Precision vs. Number of Method Covered



We mine useful java methods description from developers discussions

Supporting Software Development

Help Newcomer Program Comprehension with extraction of summaries of code elements from



StackOverflow

Term Vector Frequency in ...

stackoverflow.com/questions/12098083/term-vector-frequency-in-lucene-4-0

StackExchange ▾ sign up log in tour help ▾ careers 2.0 search

stackoverflow Questions Tags Tour Users Ask Question

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required. Take the 2-minute tour ×

Term Vector Frequency in Lucene 4.0

CAREERS 2.0 by stackoverflow + Have projects on Codeplex? Import them easily to your profile

asked 1 year ago
viewed 5845 times
active 6 months ago

2

I'm upgrading from Lucene 3.6 to Lucene 4.0-beta. In Lucene 3.x, the `IndexReader` contains a method `IndexReader.getTermFreqVectors()`, which I can use to extract the frequency of each term in a given document and field.

This method is now replaced by `IndexReader.getTermVectors()`, which returns `Terms`. How can I make use of this (or probably other methods) to extract the term frequency in a document and a field?

1

lucene

share | improve this question

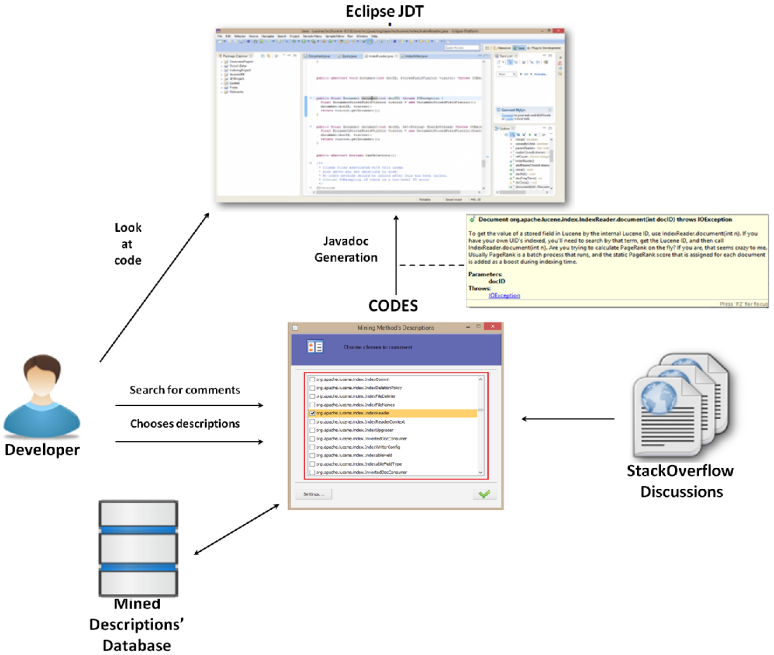
edited Aug 23 '12 at 19:09
Nathan Hughes
28.7k • 8 • 43 • 84

asked Aug 23 '12 at 18:41
mossaab
140 • 3 • 13

Related to stackoverflow.com/questions/13537126/... and stackoverflow.com/questions/1844194/... –

CODES:
Approach for Mining Method Descriptions

- Step 1: Downloading SO discussions relying on its REST interface and tracing them onto classes
- Step 2: Extracting paragraphs
- Step 3: Tracing paragraphs onto methods
  (Discards Paragraphs of discussions with 0 Votes)
- Step 4: Heuristic based Filtering
- Step 5: Similarity based Filtering



Carmine Vassallo, *Sebastiano Panichella*, Massimiliano Di Penta, Gerardo Canfora:
CODES: mining source code descriptions from developers discussions.
BEST TOOL AWARD at the 22nd International Conference on Program Comprehension (IEEE ICPC 2014)

CODES Tool:

Package Explorer

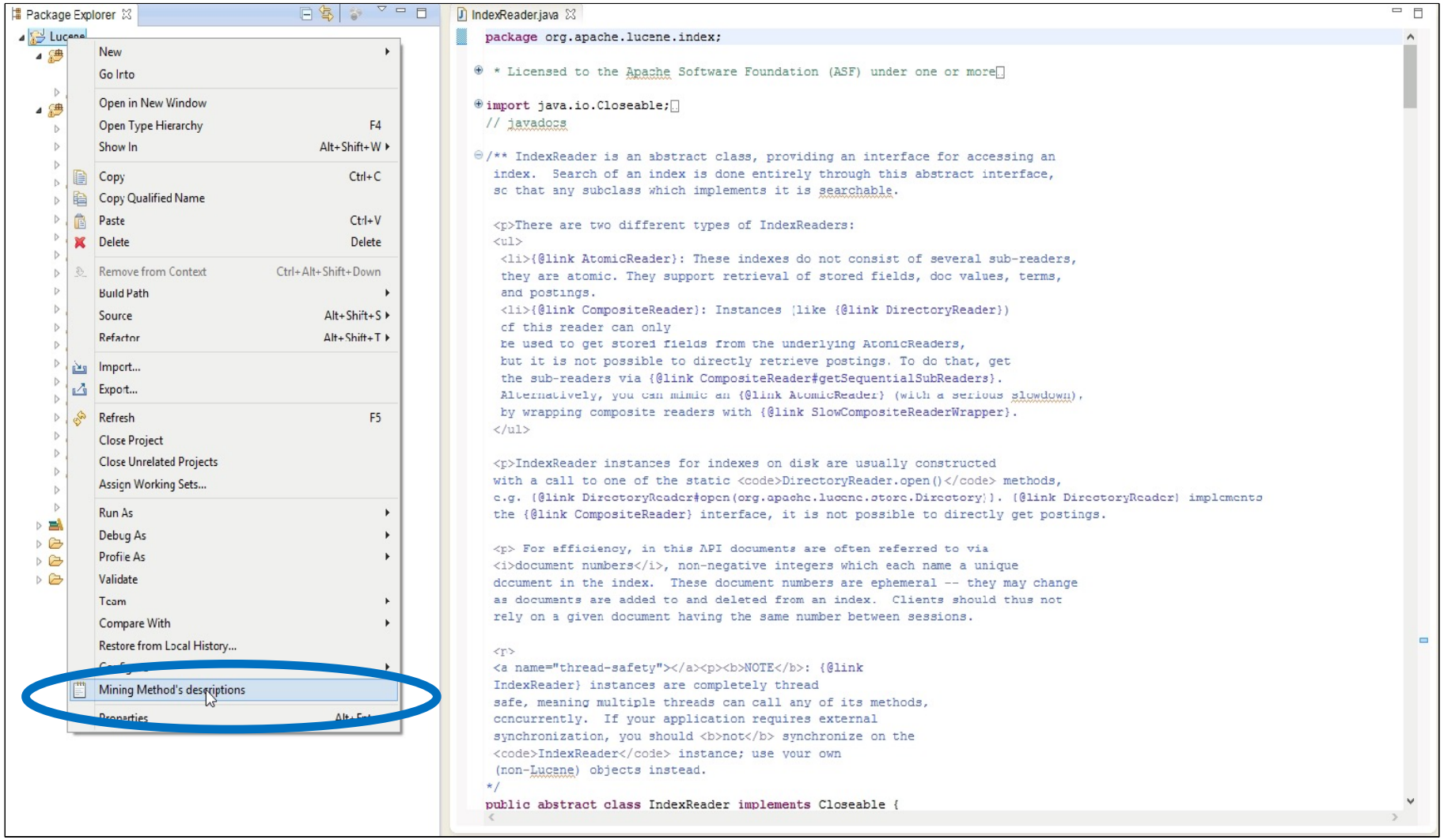
- Lucene
  - src
    - org.ap.t
    - lucene-4.1.0
  - src/lucene-4.1.0/core/src/java
    - org.apache.lucene
    - org.apache.lucene.analysis
    - org.apache.lucene.analysis.tokenattributes
    - org.apache.lucene.codecs
    - org.apache.lucene.codecs.compressing
    - org.apache.lucene.codecs.lucene3x
    - org.apache.lucene.codecs.lucene40
    - org.apache.lucene.codecs.lucene40.values
    - org.apache.lucene.codecs.lucene41
    - org.apache.lucene.codecs.perfield
    - org.apache.lucene.document
    - org.apache.lucene.index
    - org.apache.lucene.search
    - org.apache.lucene.search.payloads
    - org.apache.lucene.search.similarities
    - org.apache.lucene.search.spans
    - org.apache.lucene.store
    - org.apache.lucene.util
    - org.apache.lucene.util.automaton
    - org.apache.lucene.util.fst
    - org.apache.lucene.util.mutable
    - org.apache.lucene.util.packed
  - JRE System Library [JavaSE-1.7]
  - databasePlugin
  - oldHTML
  - XMI Plugin

IndexReader.java

```
public abstract void document(int docID, StoredFieldVisitor visitor) throws IOException;

public final Document document(int docID) throws IOException {
    final DocumentStoredFieldVisitor visitor = new DocumentStoredFieldVisitor();
    document(docID, visitor);
    return visitor.getDocument();
  }

public final Document document(int docID, Set<String> fieldsToLoad) throws IOException {
    final DocumentStoredFieldVisitor visitor = new DocumentStoredFieldVisitor(fieldsToLoad);
    document(docID, visitor);
    return visitor.getDocument();
  }

  /**
   * A custom listener that's invoked when the IndexReader
   * is closed.
   *
   * @lucene.experimental
   */
  public static interface ReaderClosedListener {
    /** Invoked when the {@link IndexReader} is closed. */
    public void onClose(IndexReader reader);
  }

  private final Set<ReaderClosedListener> readerClosedListeners =
      Collections.synchronizedSet(new LinkedHashSet<ReaderClosedListener>());

  private final Set<IndexReader> parentReaders =
      Collections.synchronizedSet(Collections.newSetFromMap(new WeakHashMap<IndexReader,Boolean>()));

  /** Expert: adds a {@link ReaderClosedListener}.  The
   * provided listener will be invoked when this reader is closed.
   *
   * @lucene.experimental */
  public final void addReaderClosedListener(ReaderClosedListener listener) {
    ensureOpen();
    readerClosedListeners.add(listener);
```

http://www.ing.unisannio.it/spanichella/pages/projects.html

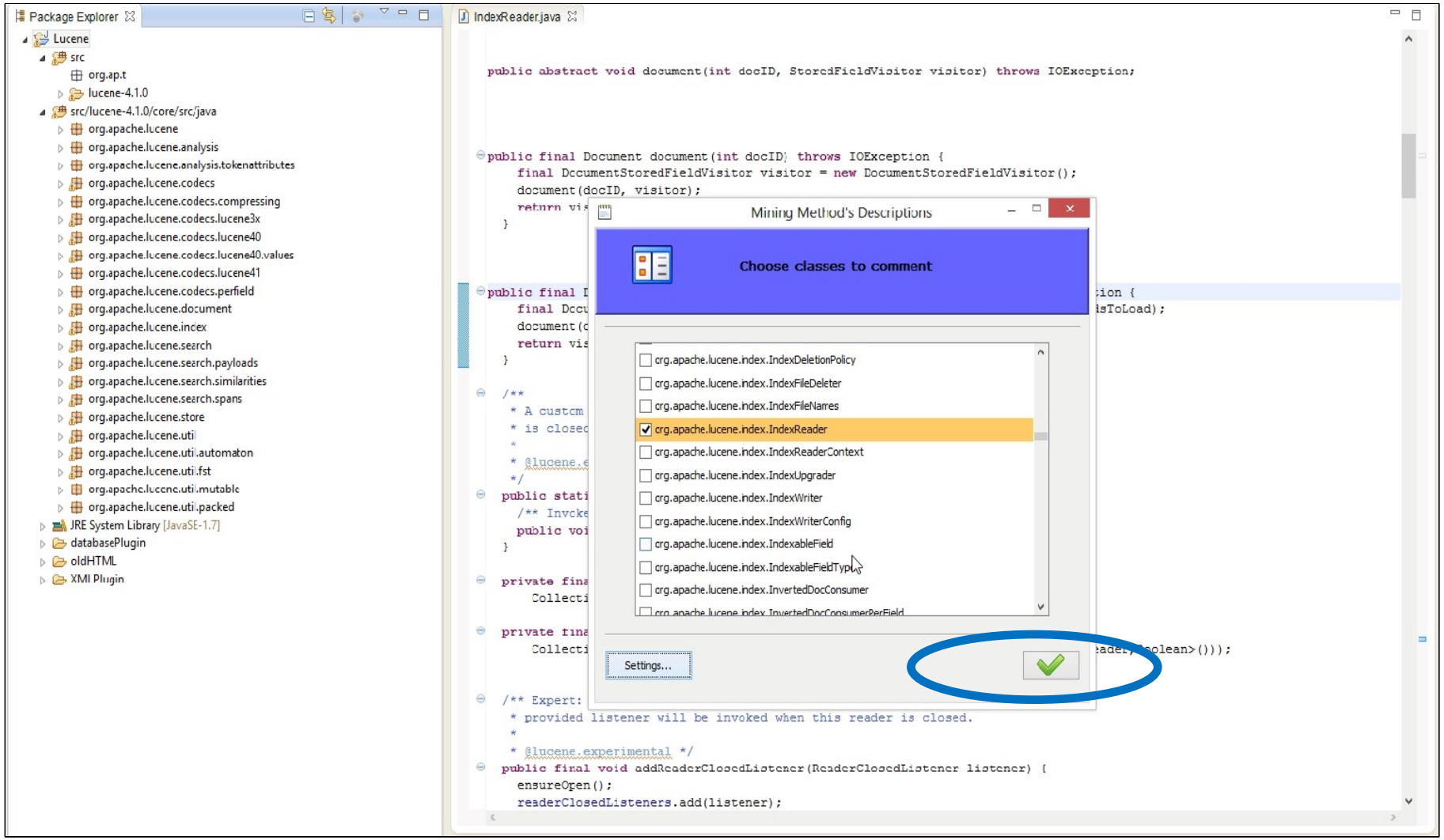
CODES Tool :


Package Explorer
Lucene
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Run As
Debug As
Profile As
Validate
Team
Compare With
Restore from Local History...
Mining Method's descriptions
IndexReader.java
package org.apache.lucene.index;
* Licensed to the Apache Software Foundation (ASF) under one or more
import java.io.Closeable;
// javadocs
/** IndexReader is an abstract class, providing an interface for accessing an
index. Search of an index is done entirely through this abstract interface,
so that any subclass which implements it is searchable.
<p>There are two different types of IndexReaders:

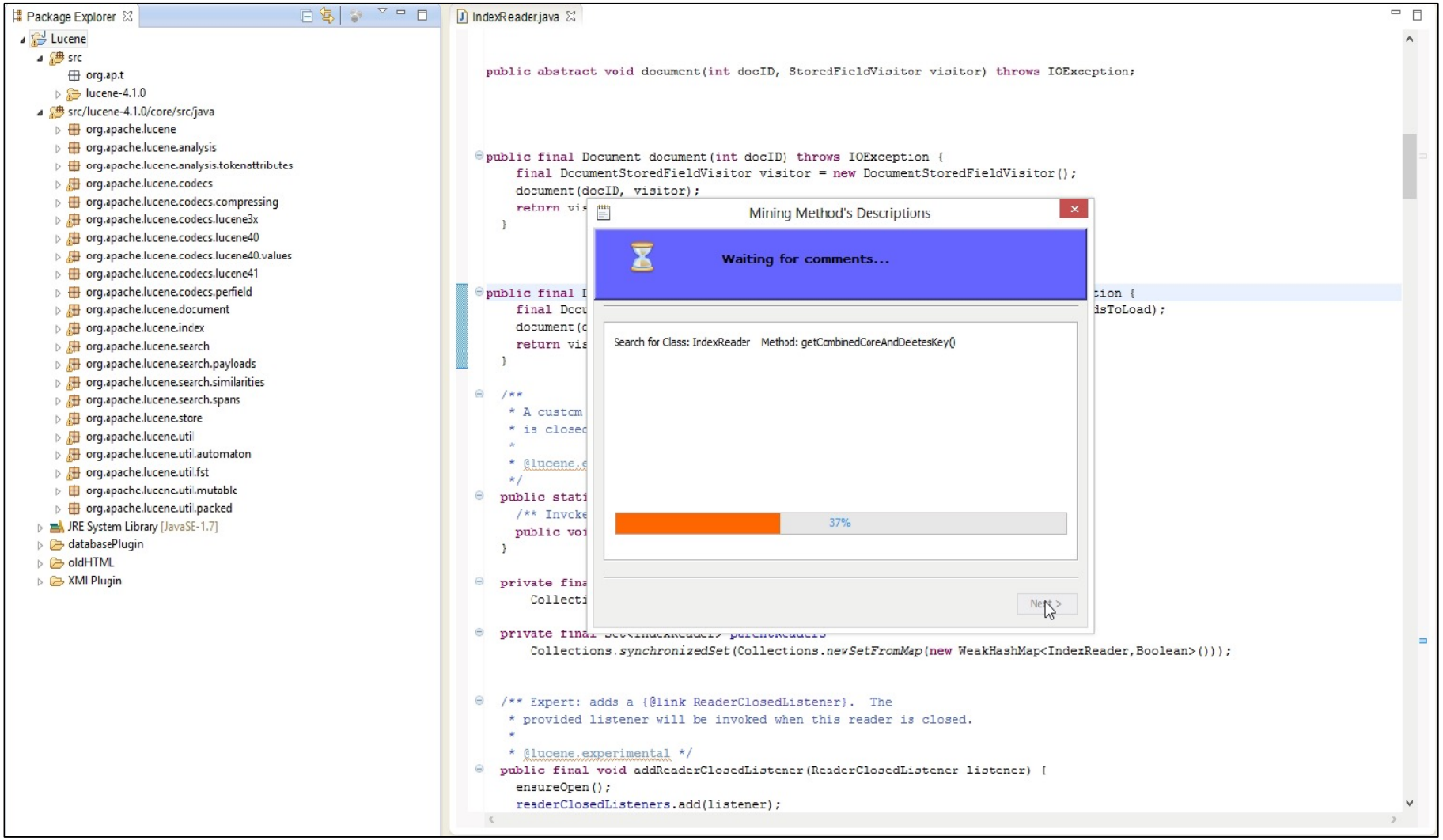
{@link AtomicReader}: These indexes do not consist of several sub-readers,
they are atomic. They support retrieval of stored fields, doc values, terms,
and postings.
{@link CompositeReader}: Instances (like {@link DirectoryReader})
of this reader can only
be used to get stored fields from the underlying AtomicReaders,
but it is not possible to directly retrieve postings. To do that, get
the sub-readers via {@link CompositeReader#getSequentialSubReaders}.
Alternatively, you can mimic an {@link AtomicReader} (with a serious slowdown),
by wrapping composite readers with {@link SlowCompositeReaderWrapper}.

<p>IndexReader instances for indexes on disk are usually constructed
with a call to one of the static <code>DirectoryReader.open()</code> methods,
e.g. {@link DirectoryReader#open(org.apache.lucene.store.Directory)}. {@link DirectoryReader} implements
the {@link CompositeReader} interface, it is not possible to directly get postings.
<p> For efficiency, in this API documents are often referred to via
document numbers, non-negative integers which each name a unique
document in the index. These document numbers are ephemeral -- they may change
as documents are added to and deleted from an index. Clients should thus not
rely on a given document having the same number between sessions.
<p>
<p>NOTE: {@link
IndexReader} instances are completely thread
safe, meaning multiple threads can call any of its methods,
concurrently. If your application requires external
synchronization, you should not synchronize on the
<code>IndexReader</code> instance; use your own
(non-Lucene) objects instead.
*/
public abstract class IndexReader implements Closeable {

CODES Tool :


Package Explorer
Lucene
src
org.ap.t
lucene-4.1.0
src/lucene-4.1.0/core/src/java
org.apache.lucene
org.apache.lucene.analysis
org.apache.lucene.analysis.tokenattributes
org.apache.lucene.codecs
org.apache.lucene.codecs.compressing
org.apache.lucene.codecs.lucene3x
org.apache.lucene.codecs.lucene40
org.apache.lucene.codecs.lucene40.values
org.apache.lucene.codecs.lucene41
org.apache.lucene.codecs.perfield
org.apache.lucene.document
org.apache.lucene.index
org.apache.lucene.search
org.apache.lucene.search.payloads
org.apache.lucene.search.similarities
org.apache.lucene.search.spans
org.apache.lucene.store
org.apache.lucene.util
org.apache.lucene.util.automaton
org.apache.lucene.util.fst
org.apache.lucene.util.mutable
org.apache.lucene.util.packed
JRE System Library [JavaSE-1.7]
databasePlugin
oldHTML
XMI Plugin
IndexReader.java
public abstract void document(int docID, StoredFieldVisitor visitor) throws IOException;
public final Document document(int docID) throws IOException {
final DocumentStoredFieldVisitor visitor = new DocumentStoredFieldVisitor();
document(docID, visitor);
Mining Method's Descriptions
Choose classes to comment
org.apache.lucene.index.IndexDeletionPolicy
org.apache.lucene.index.IndexFileDeleter
org.apache.lucene.index.IndexFileNames
org.apache.lucene.index.IndexReader
org.apache.lucene.index.IndexReaderContext
org.apache.lucene.index.IndexUpgrader
org.apache.lucene.index.IndexWriter
org.apache.lucene.index.IndexWriterConfig
org.apache.lucene.index.IndexableField
org.apache.lucene.index.InvertedDocConsumer
Settings...
/** Expert:
* provided listener will be invoked when this reader is closed.
* @lucene.experimental */
public final void addReaderClosedListener(ReaderClosedListener listener) {
ensureOpen();
readerClosedListeners.add(listener);

CODES Tool:


Package Explorer
Lucene
src
org.ap.t
lucene-4.1.0
src/lucene-4.1.0/core/src/java
org.apache.lucene
org.apache.lucene.analysis
org.apache.lucene.analysis.tokenattributes
org.apache.lucene.codecs
org.apache.lucene.codecs.compressing
org.apache.lucene.codecs.lucene3x
org.apache.lucene.codecs.lucene40
org.apache.lucene.codecs.lucene40.values
org.apache.lucene.codecs.lucene41
org.apache.lucene.codecs.perfield
org.apache.lucene.document
org.apache.lucene.index
org.apache.lucene.search
org.apache.lucene.search.payloads
org.apache.lucene.search.similarities
org.apache.lucene.search.spans
org.apache.lucene.store
org.apache.lucene.util
org.apache.lucene.util.automaton
org.apache.lucene.util.fst
org.apache.lucene.util.mutable
org.apache.lucene.util.packed
JRE System Library [JavaSE-1.7]
databasePlugin
oldHTML
XMI Plugin
IndexReader.java
public abstract void document(int docID, StoredFieldVisitor visitor) throws IOException;
public final Document document(int docID) throws IOException {
final DocumentStoredFieldVisitor visitor = new DocumentStoredFieldVisitor();
document(docID, visitor);
Mining Method's Descriptions
Waiting for comments...
Search for Class: IndexReader Method: getCombinedCoreAndDeletesKey()
37%
Next >
Collections.synchronizedSet(Collections.newSetFromMap(new WeakHashMap<IndexReader,Boolean>()));
/** Expert: adds a {@link ReaderClosedListener}. The
* provided listener will be invoked when this reader is closed.
* @lucene.experimental */
public final void addReaderClosedListener(ReaderClosedListener listener) {
ensureOpen();
readerClosedListeners.add(listener);

CODES Tool:

Package Explorer

- Lucene
  - src
    - org.ap.t
    - lucene-4.1.0
  - src/lucene-4.1.0/core/src/java
    - org.apache.lucene
    - org.apache.lucene.analysis
    - org.apache.lucene.analysis.tokenattributes
    - org.apache.lucene.codecs
    - org.apache.lucene.codecs.compressing
    - org.apache.lucene.codecs.lucene3x
    - org.apache.lucene.codecs.lucene40
    - org.apache.lucene.codecs.lucene40.values
    - org.apache.lucene.codecs.lucene41
    - org.apache.lucene.codecs.perfield
    - org.apache.lucene.document
    - org.apache.lucene.index
    - org.apache.lucene.search
    - org.apache.lucene.search.payloads
    - org.apache.lucene.search.similarities
    - org.apache.lucene.search.spans
    - org.apache.lucene.store
    - org.apache.lucene.util
    - org.apache.lucene.util.automaton
    - org.apache.lucene.util.fst
    - org.apache.lucene.util.mutable
    - org.apache.lucene.util.packed
  - JRE System Library [JavaSE-1.7]
  - databasePlugin
  - oldHTML
  - XMI Plugin

IndexReader.java

```
public abstract void document(int docID, StoredFieldVisitor visitor) throws IOException;

public final Document document(int docID) throws IOException {
    final DocumentStoredFieldVisitor visitor = new DocumentStoredFieldVisitor();
    document(docID, visitor);
    return vis
  }

public final I                                                    tion {
    final Docu                                                    dsToLoad);
    document(c
    return vis
  }

  /**
   * A custom
   * is closed
   *
   * @lucene.e
   */
  public stati
    /** Invoke
    public voi
  }

  private fina
      Collecti

  private final Set<IndexReader> parentReaders
      Collections.synchronizedSet(Collections.newSetFromMap(new WeakHashMap<IndexReader,Boolean>()));

  /** Expert: adds a {@link ReaderClosedListener}.  The
   * provided listener will be invoked when this reader is closed.
   *
   * @lucene.experimental */
  public final void addReaderClosedListener(ReaderClosedListener listener) {
    ensureOpen();
    readerClosedListeners.add(listener);
```

Mining Method's Descriptions

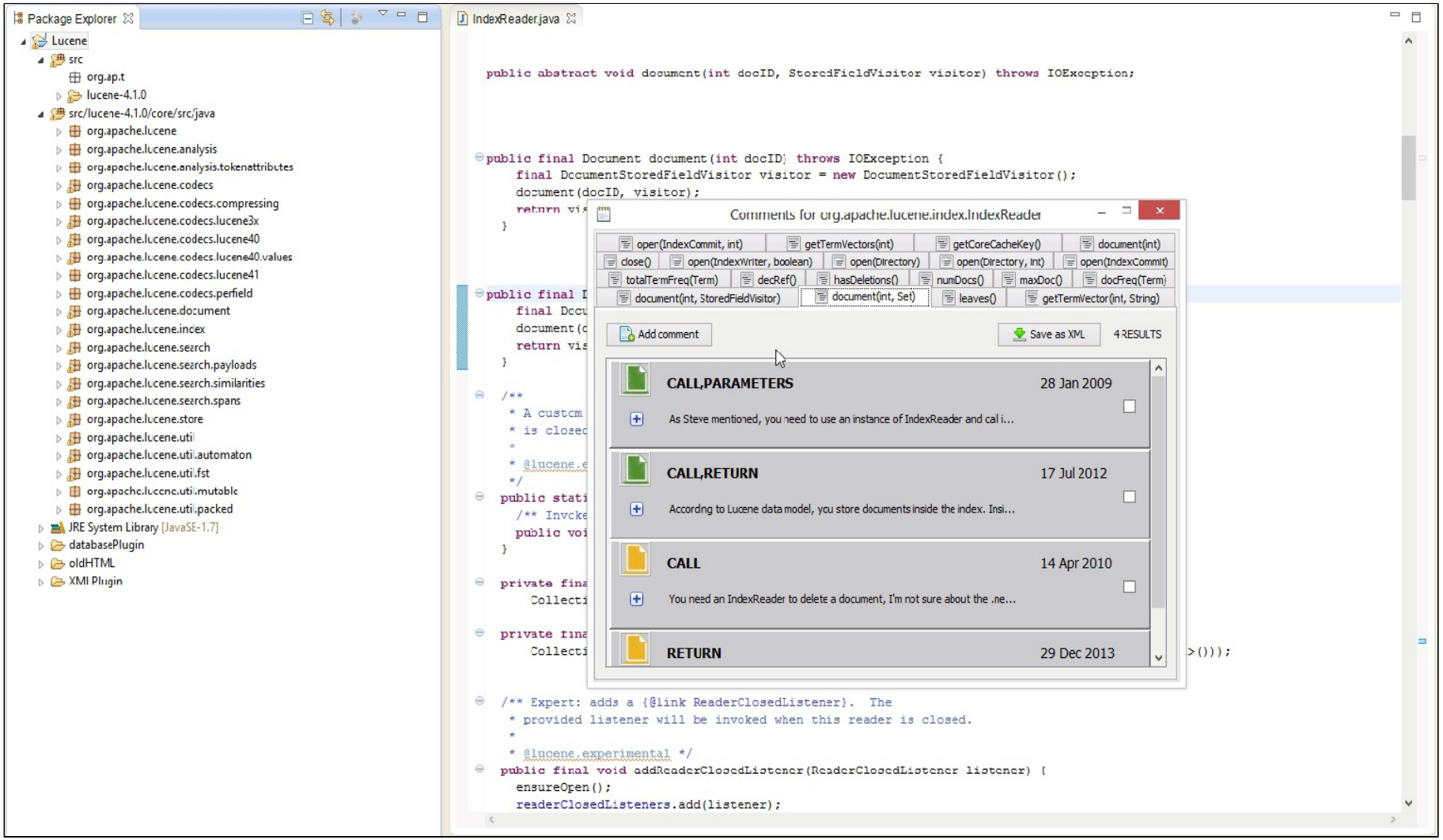
Waiting for comments...

Search completed for IndexReader

100%

Next >

CODES Tool:


Package Explorer
Lucene
src
org.ap.t
lucene-4.1.0
src/lucene-4.1.0/core/src/java
org.apache.lucene
org.apache.lucene.analysis
org.apache.lucene.analysis.tokenattributes
org.apache.lucene.codecs
org.apache.lucene.codecs.compressing
org.apache.lucene.codecs.lucene3x
org.apache.lucene.codecs.lucene40
org.apache.lucene.codecs.lucene40.values
org.apache.lucene.codecs.lucene41
org.apache.lucene.codecs.perfield
org.apache.lucene.document
org.apache.lucene.index
org.apache.lucene.search
org.apache.lucene.search.payloads
org.apache.lucene.search.similarities
org.apache.lucene.search.spans
org.apache.lucene.store
org.apache.lucene.uti
org.apache.lucene.uti.automaton
org.apache.lucene.uti.fst
org.apache.lucene.uti.mutable
org.apache.lucene.uti.packed
JRE System Library [JavaSE-1.7]
databasePlugin
oldHTML
XMI Plugin
IndexReader.java
public abstract void document(int docID, StoredFieldVisitor visitor) throws IOException;
public final Document document(int docID) throws IOException {
final DocumentStoredFieldVisitor visitor = new DocumentStoredFieldVisitor();
document(docID, visitor);
Comments for org.apache.lucene.index.IndexReader
oper(IndexCommit, int)
getTermVectors(int)
getCoreCacheKey()
document(int)
close()
open(IndexWriter, boolean)
open(Directory)
open(Directory, int)
open(IndexCommit)
totalTermFreq(Term)
decRef()
hasDeletions()
numDocs()
maxDoc()
docFreq(Term)
document(int, StoredFieldVisitor)
document(int, Set)
leaves()
getTermVector(int, String)
Add comment
Save as XML
4 RESULTS
CALL,PARAMETERS
28 Jan 2009
As Steve mentioned, you need to use an instance of IndexReader and cal i...
CALL,RETURN
17 Jul 2012
Accordng to Lucene data model, ycu store documents inside the index. Insi...
CALL
14 Apr 2010
You need an IndexReader to delete a document, I'm not sure about the .ne...
RETURN
29 Dec 2013
/** Expert: adds a {@link ReaderClosedListener}. The
* provided listener will be invoked when this reader is closed.
* @lucene.experimental */
public final void addReaderClosedListener(ReaderClosedListener listener) {
ensureOpen();
readerClosedListeners.add(listener);

CODES Tool:

Package Explorer

- Lucene
  - src
    - org.ap.t
    - lucene-4.1.0
  - src/lucene-4.1.0/core/src/java
    - org.apache.lucene
    - org.apache.lucene.analysis
    - org.apache.lucene.analysis.tokenattributes
    - org.apache.lucene.codecs
    - org.apache.lucene.codecs.compressing
    - org.apache.lucene.codecs.lucene3x
    - org.apache.lucene.codecs.lucene40
    - org.apache.lucene.codecs.lucene40.values
    - org.apache.lucene.codecs.lucene41
    - org.apache.lucene.codecs.perfield
    - org.apache.lucene.document
    - org.apache.lucene.index
    - org.apache.lucene.search
    - org.apache.lucene.search.payloads
    - org.apache.lucene.search.similarities
    - org.apache.lucene.search.spans
    - org.apache.lucene.store
    - org.apache.lucene.util
    - org.apache.lucene.util.automaton
    - org.apache.lucene.util.fst
    - org.apache.lucene.util.mutable
    - org.apache.lucene.util.packed
  - JRE System Library [JavaSE-1.7]
  - databasePlugin
  - oldHTML
  - XMI Plugin

IndexReader.java

```
public abstract void document(int docID, StoredFieldVisitor visitor) throws IOException;

public final Document document(int docID) throws IOException {
    final DocumentStoredFieldVisitor visitor = new DocumentStoredFieldVisitor();
    document(docID, visitor);
    return vis
}

public final D
    final Docu
    document(d
    return vis
}

/**
 * A custom
 * is closed
 *
 * @lucene.e
 */
public stati
    /** Invoke
    public voi
}

private fina
    Collecti

private fina
    Collecti

/** Expert: adds a {@link ReaderClosedListener}.  The
 * provided listener will be invoked when this reader is closed.
 *
 * @lucene.experimental */
public final void addReaderClosedListener(ReaderClosedListener listener) {
    ensureOpen();
    readerClosedListeners.add(listener);
```

Comments for org.apache.lucene.index.IndexReader

oper(IndexCommit, int) | getTermVectors(int) | getCoreCacheKey() | document(int)
close() | open(IndexWriter, boolean) | open(Directory) | open(Directory, int) | open(IndexCommit)
totalTermFreq(Term) | decRef() | hasDeletions() | numDocs() | maxDoc() | docFreq(Term)
document(int, StoredFieldVisitor) | document(int, Set) | leaves() | getTermVector(int, String)

Add comment | Other | Save as XML | 4 RESULTS

CALL,PARAMETERS 28 Jan 2009

IndexReader call its DeleteDocuments method. DeleteDocuments accepts eit...

IndexReader call its DeleteDocuments method. DeleteDocuments accepts either an instance of a Term object or Lucene's internal id of the document (it is generally not recommended to use the internal id as it can and will change as Lucene merges segments).

CALL,RETURN 17 Jul 2012

According to Lucene data model, you store documents inside the index. Insi...

CODES Tool:

Package Explorer

- Lucene
  - src
    - org.ap.t
    - lucene-4.1.0
  - src/lucene-4.1.0/core/src/java
    - org.apache.lucene
    - org.apache.lucene.analysis
    - org.apache.lucene.analysis.tokenattributes
    - org.apache.lucene.codecs
    - org.apache.lucene.codecs.compressing
    - org.apache.lucene.codecs.lucene3x
    - org.apache.lucene.codecs.lucene40
    - org.apache.lucene.codecs.lucene40.values
    - org.apache.lucene.codecs.lucene41
    - org.apache.lucene.codecs.perfield
    - org.apache.lucene.document
    - org.apache.lucene.index
    - org.apache.lucene.search
    - org.apache.lucene.search.payloads
    - org.apache.lucene.search.similarities
    - org.apache.lucene.search.spans
    - org.apache.lucene.store
    - org.apache.lucene.util
    - org.apache.lucene.util.automaton
    - org.apache.lucene.util.fst
    - org.apache.lucene.util.mutable
    - org.apache.lucene.util.packed
  - JRE System Library [JavaSE-1.7]
  - databasePlugin
  - oldHTML
  - XMI Plugin

IndexReader.java

```
public abstract void document(int docID, StoredFieldVisitor visitor) throws IOException;

public final Document document(int docID) throws IOException {
    final DocumentStoredFieldVisitor visitor = new DocumentStoredFieldVisitor();
    document(docID, visitor);
    return vis
}

public final D
    final Docu
    document(d
    return vis
}

/**
 * A custom
 * is closed
 *
 * @lucene.e
 */
public stati
    /** Invoke
    public voi
}

private fina
    Collecti

private fina
    Collecti

/** Expert: adds a {@link ReaderClosedListener}.  The
 * provided listener will be invoked when this reader is closed.
 *
 * @lucene.experimental */
public final void addReaderClosedListener(ReaderClosedListener listener) {
    ensureOpen();
    readerClosedListeners.add(listener);
```

Insert this comment...

IndexReader call its DeleteDocuments method. DeleteDocuments accepts either an instance of a Term object or Lucene's internal id of the document (it is generally not recommended to use the internal id as it can and will change as Lucene merges segments).

Undo | Save

IndexReader

reCacheKey() | document(int) | (Directory, int) | open(IndexCommit) | maxDoc() | docFreq(Term) | es() | getTermVector(int, String)

Save as XML | 4 RESULTS

28 Jan 2009

ts eit...

eDocuments accepts eit
nal id of the document
nal id as it can and w

CALL,RETURN | 17 Jul 2012

According to Lucene data model, you store documents inside the index. Insi...

CODES Tool:

Package Explorer

- Lucene
  - src
    - org.ap.t
    - lucene-4.1.0
  - src/lucene-4.1.0/core/src/java
    - org.apache.lucene
    - org.apache.lucene.analysis
    - org.apache.lucene.analysis.tokenattributes
    - org.apache.lucene.codecs
    - org.apache.lucene.codecs.compressing
    - org.apache.lucene.codecs.lucene3x
    - org.apache.lucene.codecs.lucene40
    - org.apache.lucene.codecs.lucene40.values
    - org.apache.lucene.codecs.lucene41
    - org.apache.lucene.codecs.perfield
    - org.apache.lucene.document
    - org.apache.lucene.index
    - org.apache.lucene.search
    - org.apache.lucene.search.payloads
    - org.apache.lucene.search.similarities
    - org.apache.lucene.search.spans
    - org.apache.lucene.store
    - org.apache.lucene.util
    - org.apache.lucene.util.automaton
    - org.apache.lucene.util.fst
    - org.apache.lucene.util.mutable
    - org.apache.lucene.util.packed
  - JRE System Library [JavaSE-1.7]
  - databasePlugin
  - oldHTML
  - XMI Plugin

*IndexReader.java

```
    }

  public abstract void document(int docID, StoredFieldVisitor visitor) throws IOException;

  public final Document document(int docID) throws IOException {
      final DocumentStoredFieldVisitor visitor = new DocumentStoredFieldVisitor();
      document(docID, visitor);
      return visitor.getDocument();
    }

  /**
   * IndexReader call its DeleteDocuments method. DeleteDocuments accepts either
   * an instance of a Term object or Lucene's internal id of the document (it is
   * generally not recommended to use the internal id as it can and will change
   * as Lucene merges segments).
   */
  public final Document document(int docID, Set<String> fieldsToLoad) throws IOException {
      final DocumentStoredFieldVisitor visitor = new DocumentStoredFieldVisitor(fieldsToLoad);
      document(docID, visitor);
      return visitor.getDocument();
    }

  /**
   * A custom listener that's invoked when the IndexReader
   * is closed.
   *
   * @lucene.experimental
   */
  public static interface ReaderClosedListener {
    /** Invoked when the {@link IndexReader} is closed. */
    public void onClose(IndexReader reader);
  }

  private final Set<ReaderClosedListener> readerClosedListeners =
      Collections.synchronizedSet(new LinkedHashSet<ReaderClosedListener>());

  private final Set<IndexReader> parentReaders =
      Collections.synchronizedSet(Collections.newSetFromMap(new WeakHashMap<IndexReader,Boolean>()));
```

PART III

Recommenders

Summary

1) YODA make it possible to recommend mentors with a precision higher than 67%

2) CODES identifies relevant descriptions with a precision higher than 79%

3) Combining Mails and Issues improve recommenders' performance

Future Work and Conclusion

Future work…

Performing a survey asking to developers to validate of the social links identified by analyzing different communication channels.

Improve Existing Recommenders

Improve the mentor recommender (YODA) by considering factors able to better capture the technical skills of mentors.

Improve CODES increasing the precision and coverage as high as possible reducing the percentage of false positives. Include a better classification of discussion content using of natural language parsers.

New Recommenders

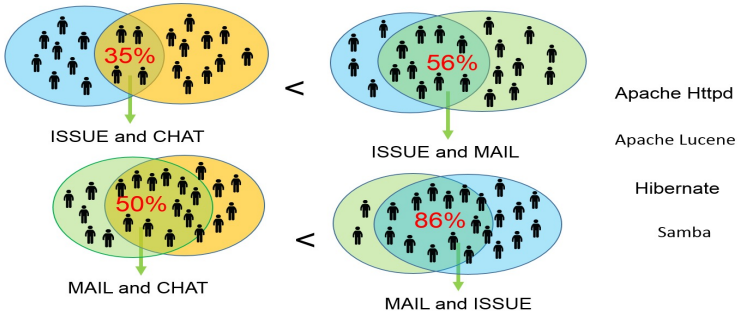
We will aim at building recommenders to help newcomer in the choice of appropriate patterns to navigate software documentation during maintenance tasks.

Building better recommenders for software re-modularization or refactoring based on social interaction between developers.

G. Bavota, Sebastiano Panichella, N. Tsantalis, M. Di Penta, R. Oliveto G. Canfora
Recommending Refactorings based on Team Co-Maintenance Patterns.
The 29th International Conference on Automated Software Engineering (ASE 2014)

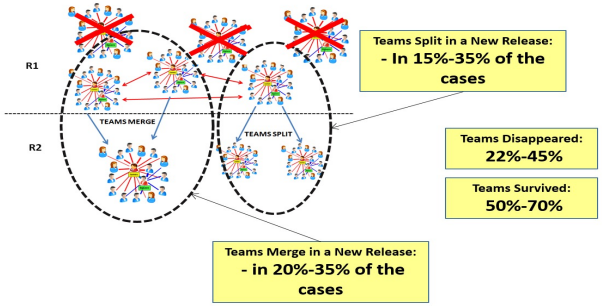
PART I

Developers overlap between Different Sources



Projects with a very high number of contributors in the 'chat' channel.
Thus, with an higher overlap between the 'mail' and 'issue' channels.

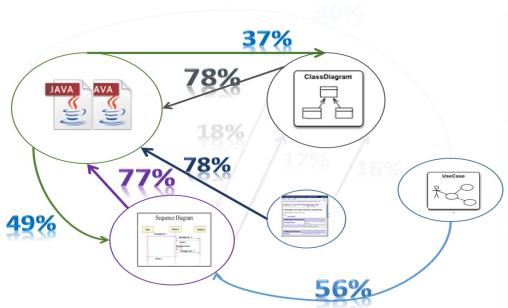
How do Emerging Collaborations Change across Software Releases?



36

PART II

Transition Graph between kinds of Software Artifacts

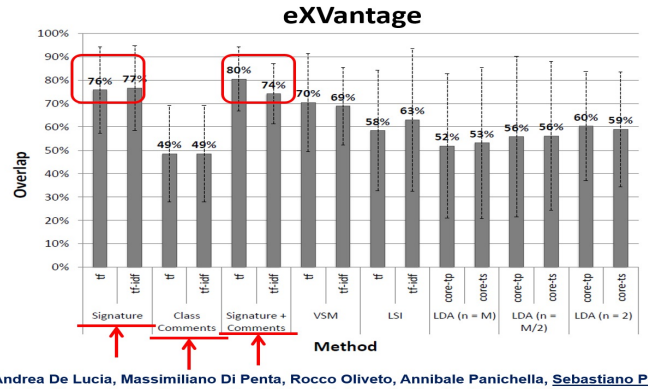


1) From Source Code participants in most cases "go back" to Sequence and Class Diagrams

2) From Sequence and Class Diagrams participants in most cases "go back" to Source Code

3) Starting from a Use Case, participants go ahead reading Sequence Diagrams. Only after, they reading and writing Source Code

Comparison of Different Labeling Techniques



Andrea De Lucia, Massimiliano Di Penta, Rocco Oliveto, Annibale Panichella, Sebastiano Panichella : Labeling Source Code with Information Retrieval Methods: An Empirical Study. EMSE 20143.

PART III

YODA
(Young and newcOmer Developer Assistant)

Approach for Mentors Identification in Open Source Projects

1) Find Past Successful Mentors

2) Suggest Mentors Having Specific Skills

65