

Impact of Developer Turnover on Quality in Open-Source Software

Matthieu Foucault^{*}, Marc Palyart[†], Xavier Blanc^{*},
Gail C. Murphy[†], Jean-Rémy Falleri^{*}

^{*}University of Bordeaux, France

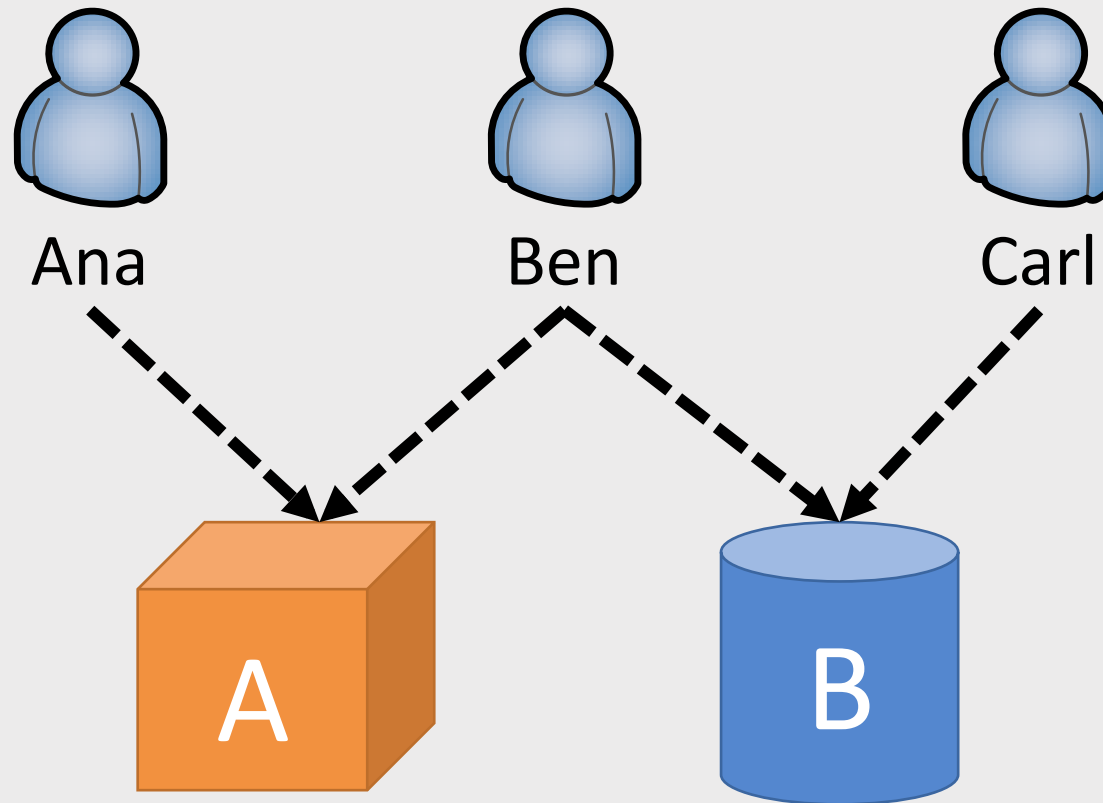
[†]University of British Columbia, Canada

Outline

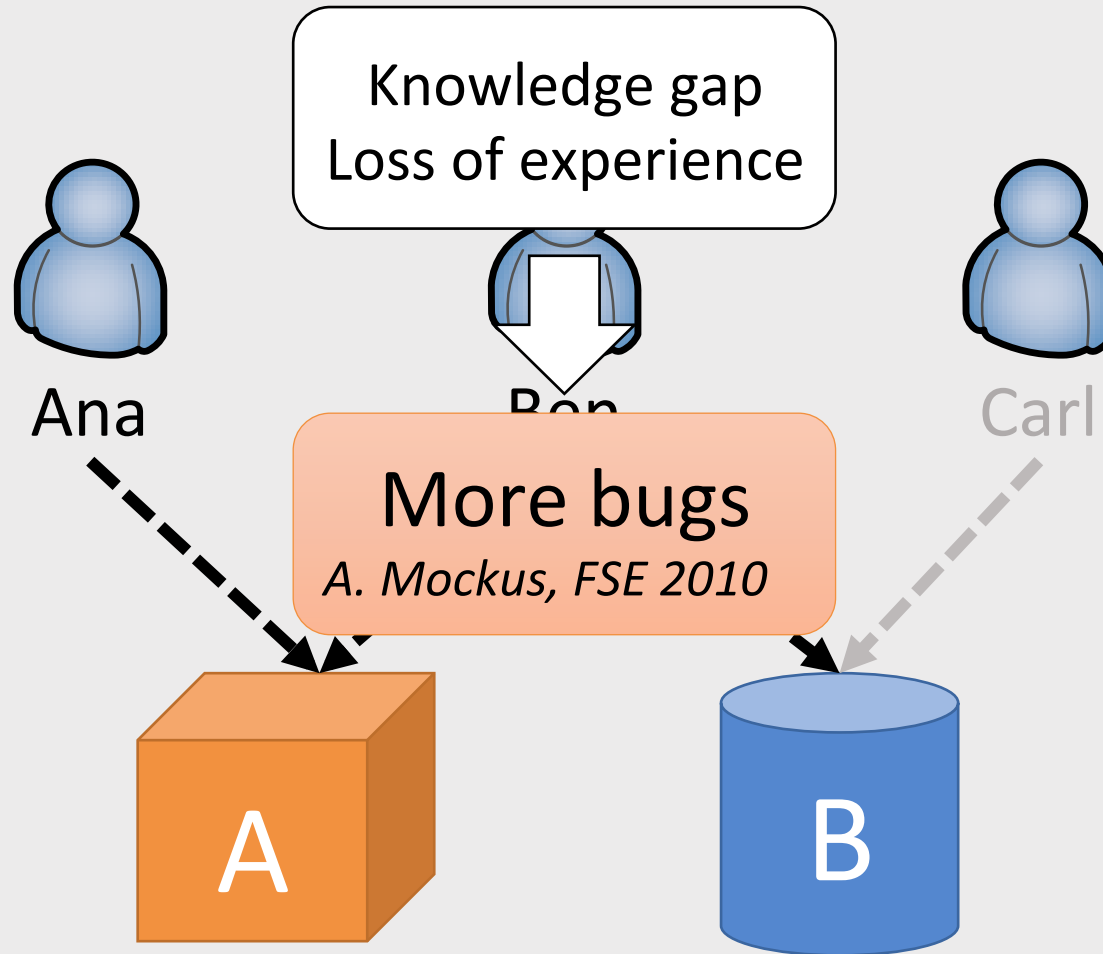
- Turnover theories
- Measure turnover
- Turnover patterns
- Relationship with quality
- Replicate

Turnover Theories

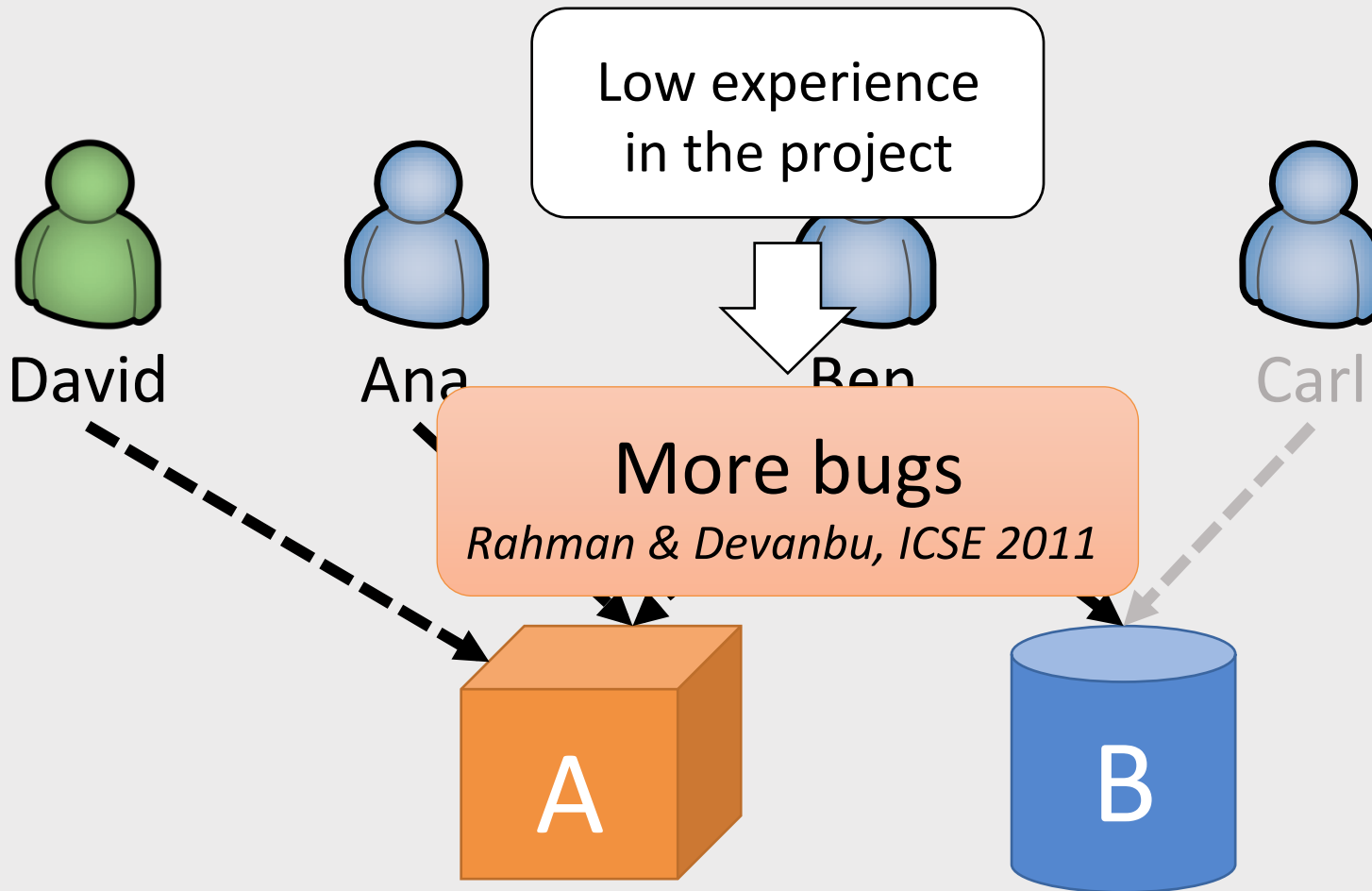
3 Developers, 2 Modules



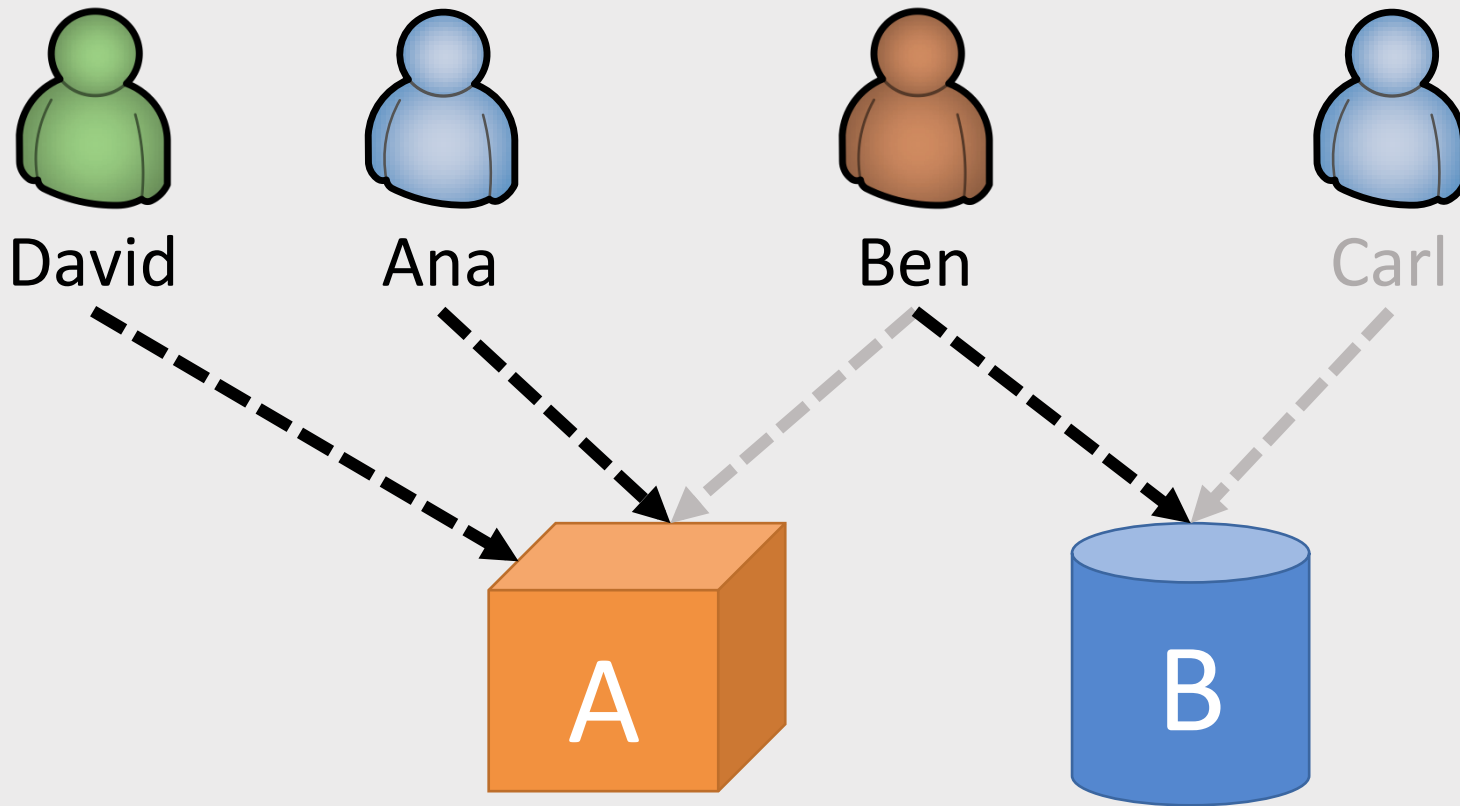
External Leavers



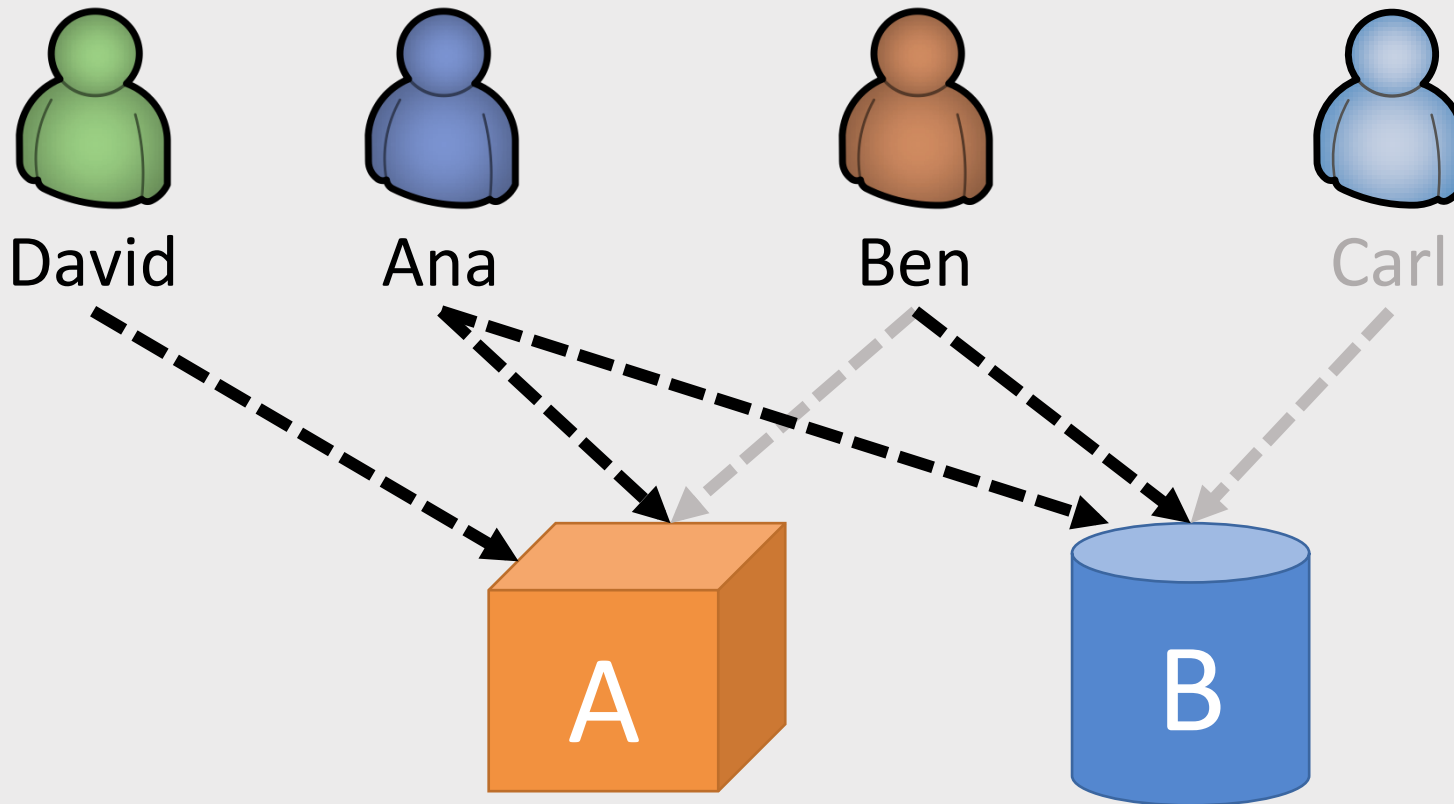
External Newcomers



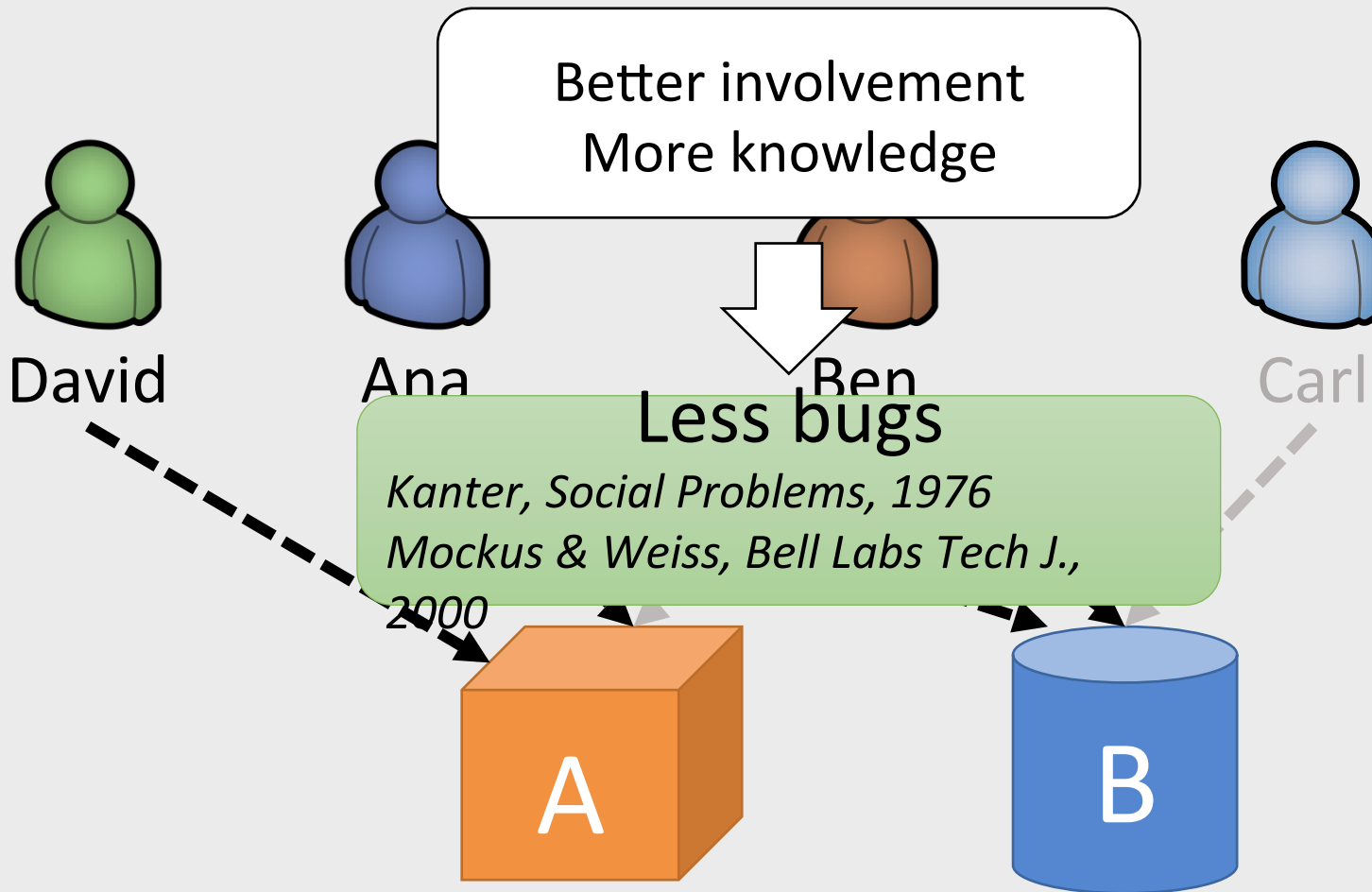
Internal Leavers



Internal Newcomers



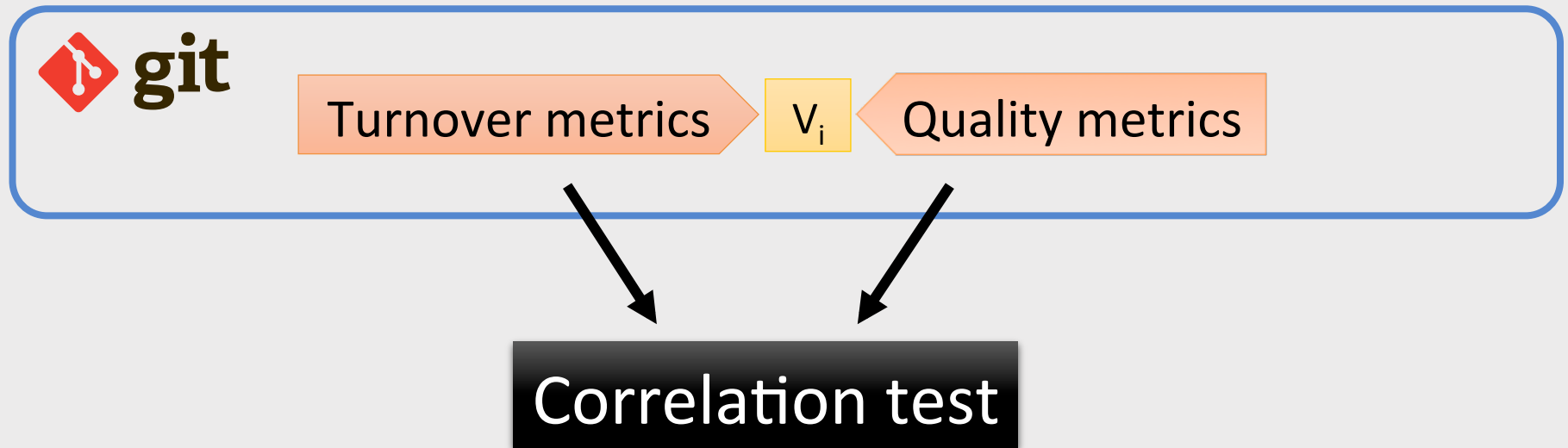
Internal Turnover



Research Questions

- 1) What are the trends of external turnover at the project level?
- 2) What are the patterns of external and internal turnover at the module level?
- 3) What is the relationship between developer turnover and software quality?

Methodology Overview



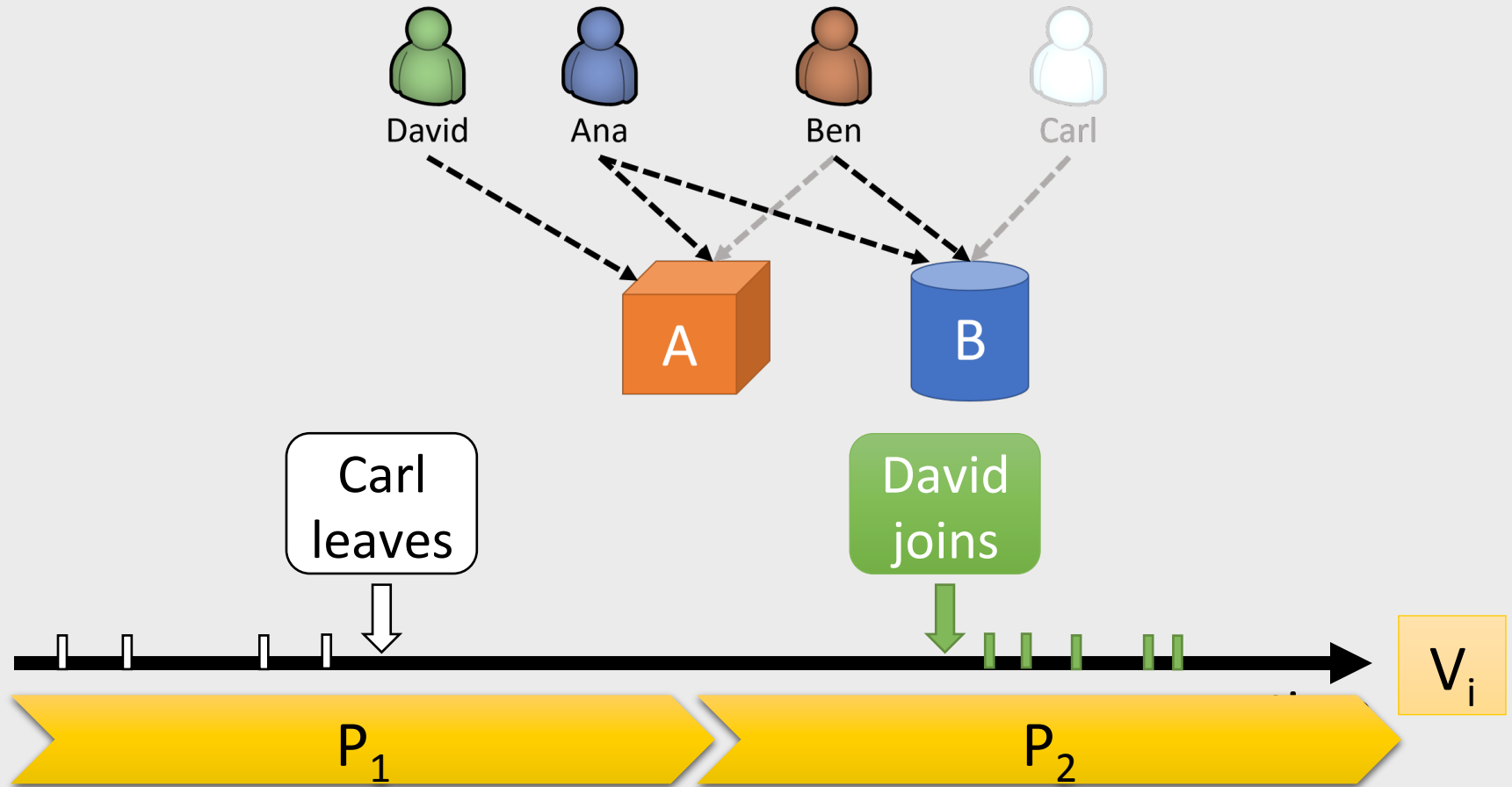
Dataset - Five open-source projects



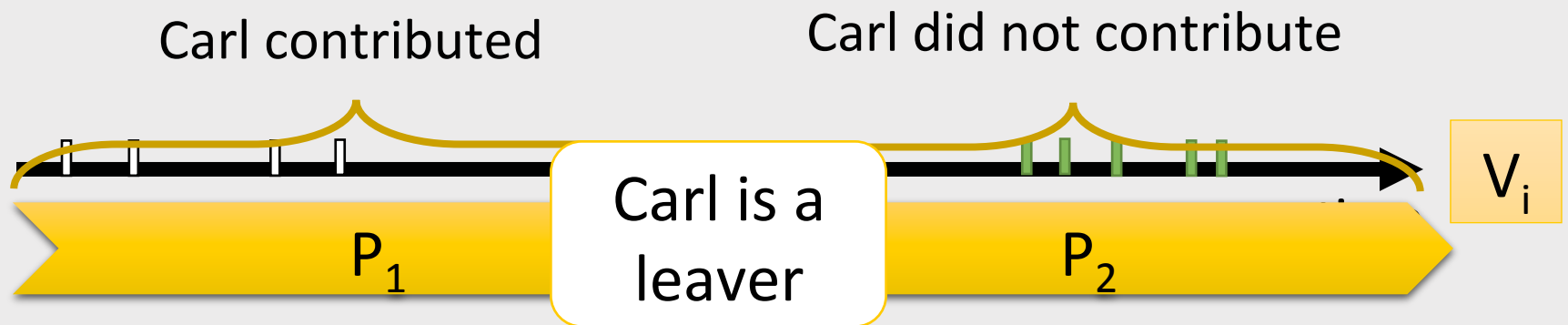
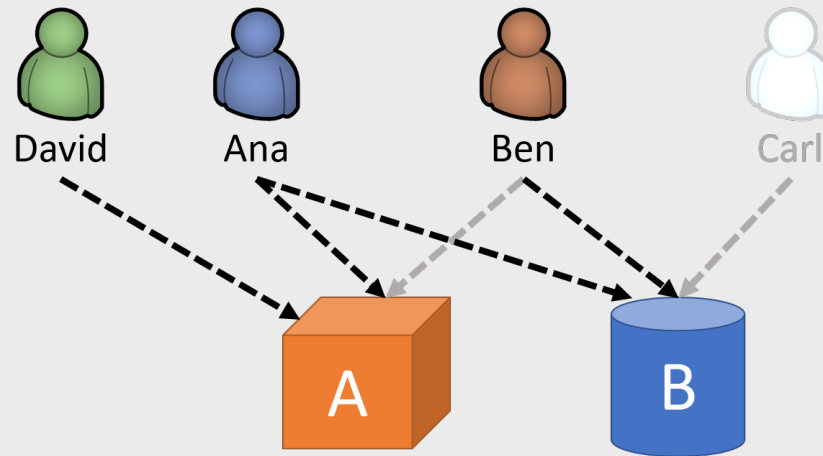
- Four programming languages (Java, JavaScript, Ruby, Python)
- From 5 to 80 kLoC
- We select **one release** in each project

How to Measure Turnover?

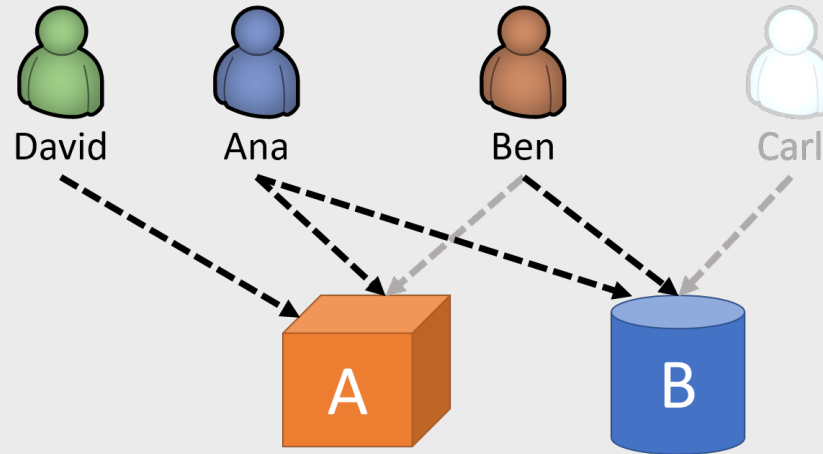
Turnover Metrics – Background



Compute Leavers

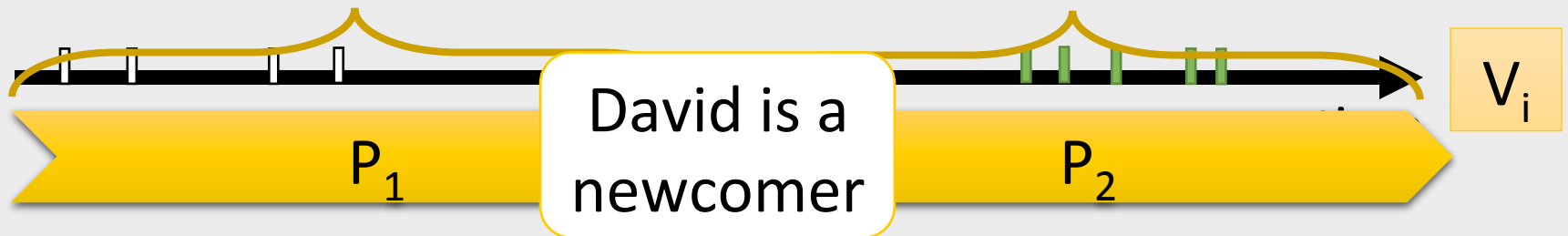


Compute Newcomers

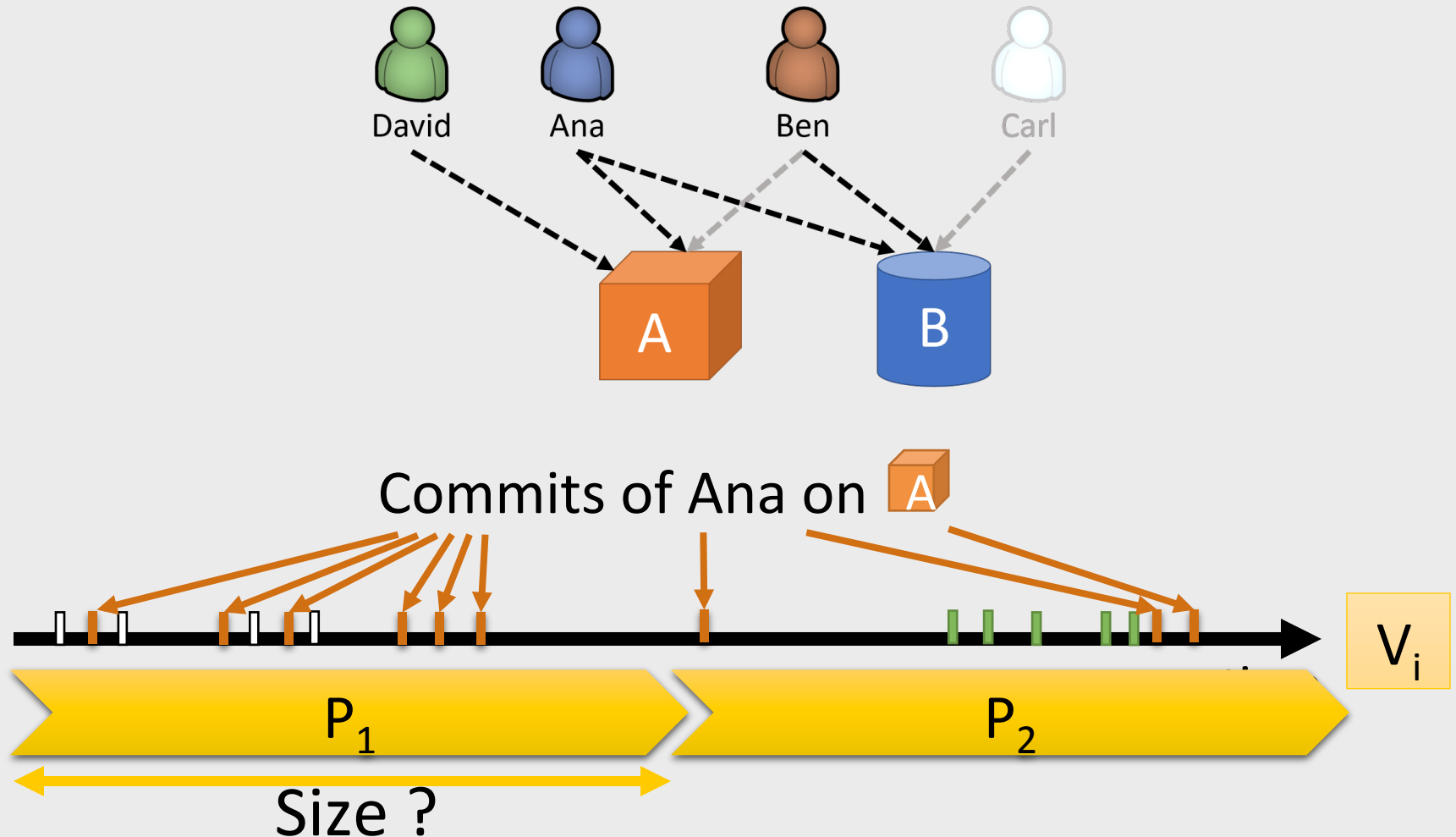


David did not contribute

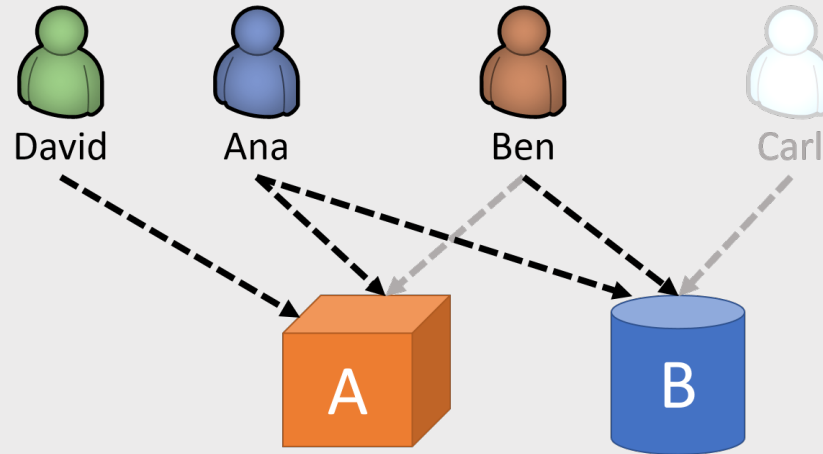
David contributed



Problem 1: Choose the Period Size



Too Small

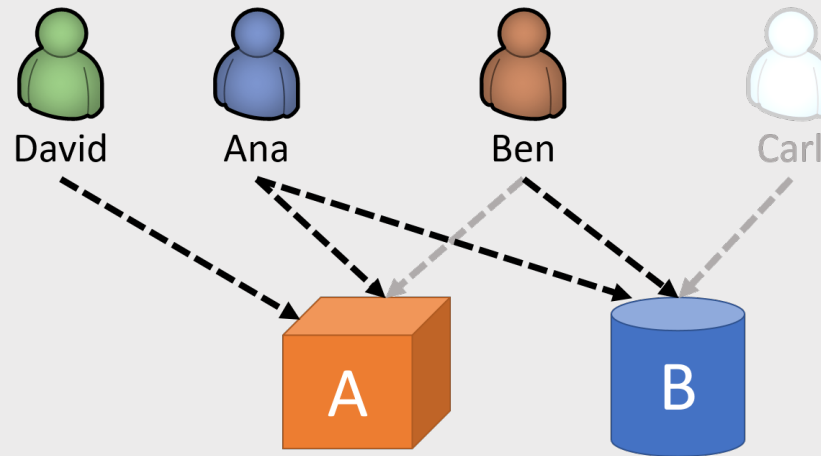


Carl is not a contributor

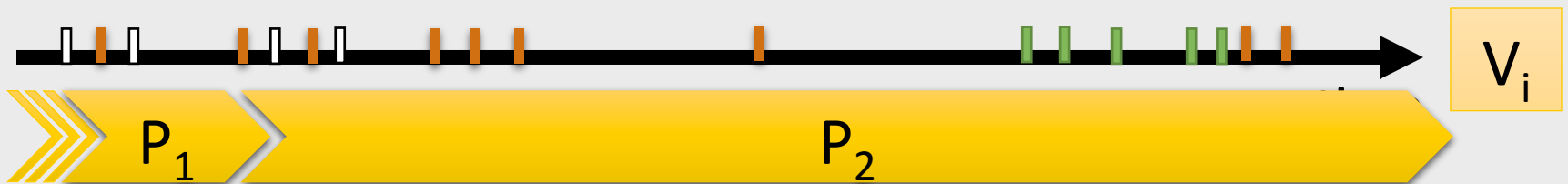
David is not an external newcomer



Too Large



Carl is not a
leaver



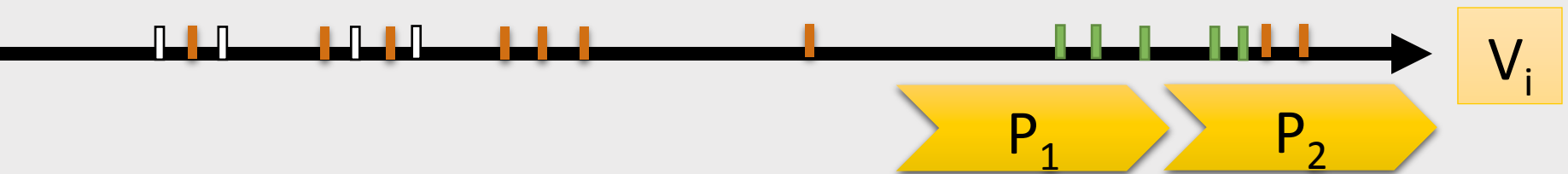
Obsolete contributions may be considered

Period Selection

Find the smallest period size where the “limited” visibility does not impact the sets of turnover actors.

For newcomers, we compare actors obtained with

Limited Visibility: $|P_1| == |P_2|$



Period Selection - Newcomers

Find the smallest period size where the “limited” visibility does not impact the sets of turnover actors.

To the ones obtained with

Full Visibility: P_1 starts with first commit

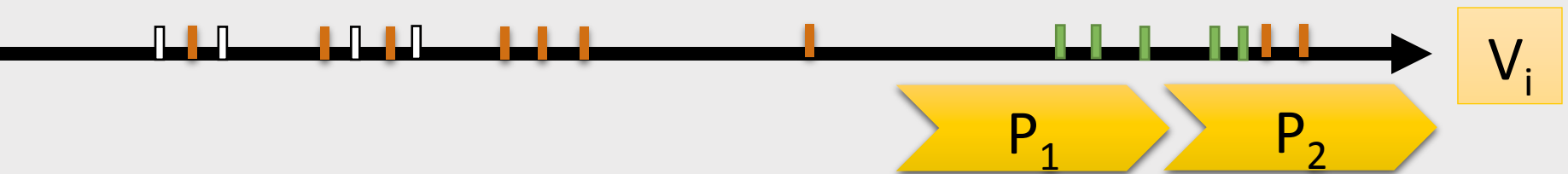


Period Selection - Leavers

Find the smallest period size where the “limited” visibility does not impact the sets of turnover actors.

For **leavers**, we compare actors obtained with

Limited Visibility: $|P_1| == |P_2|$

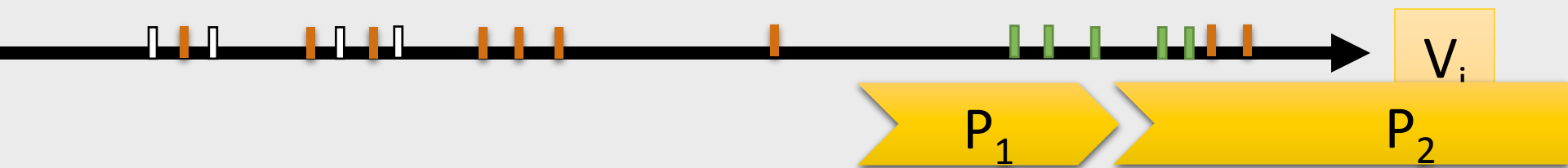


Period Selection - Leavers

Find the smallest period size where the “limited” visibility does not impact the sets of turnover actors.

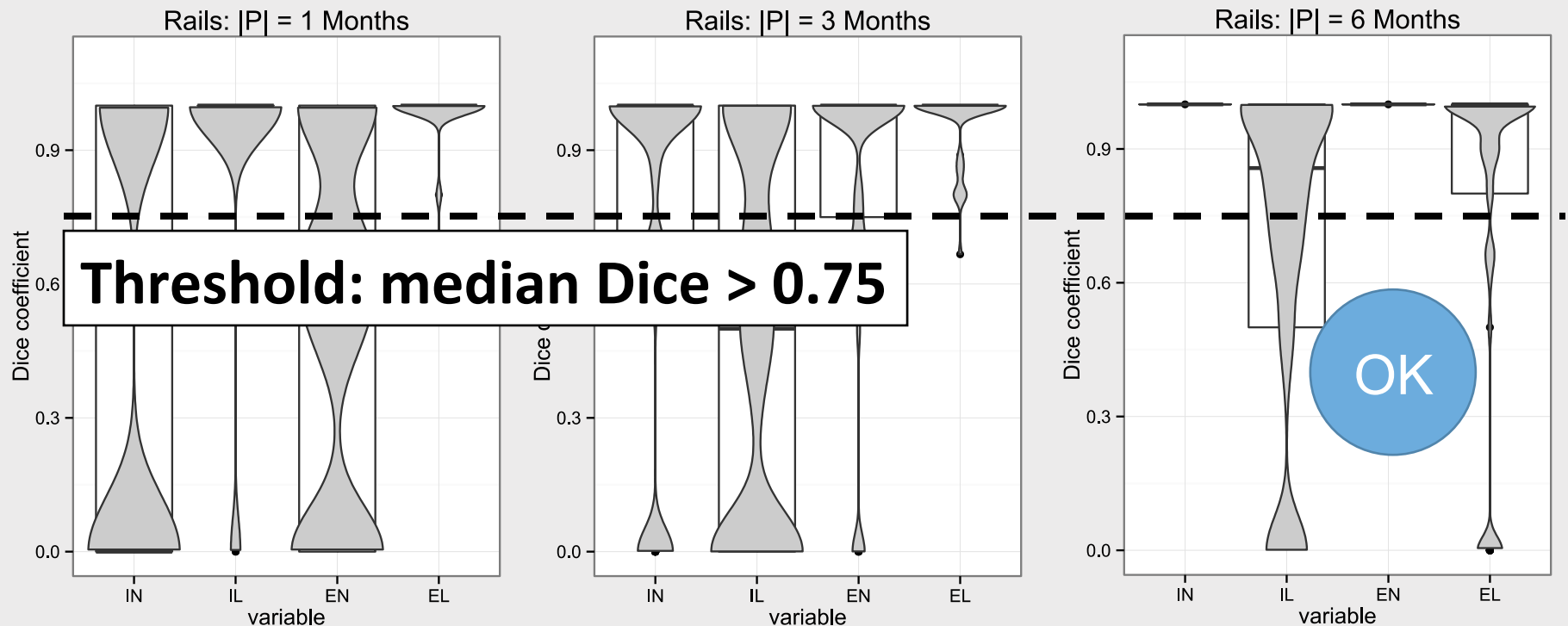
To the ones obtained with

Full Visibility: P_2 ends with last commit

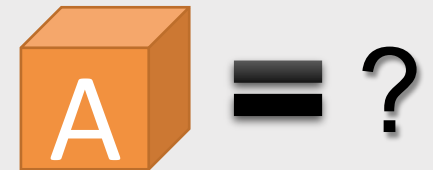
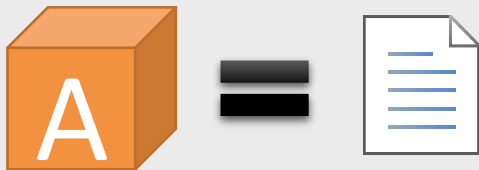
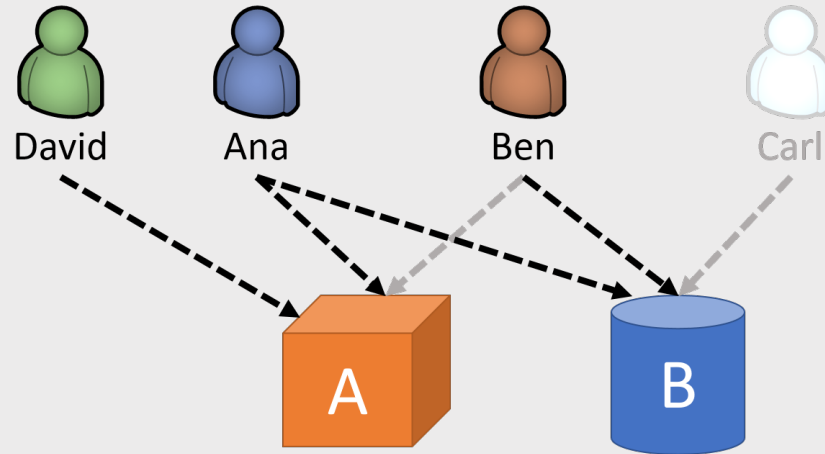


Period Selection

We use the Dice coefficient to compare the sets of internal/external leavers/newcomers obtained for each module, with limited and full visibility.



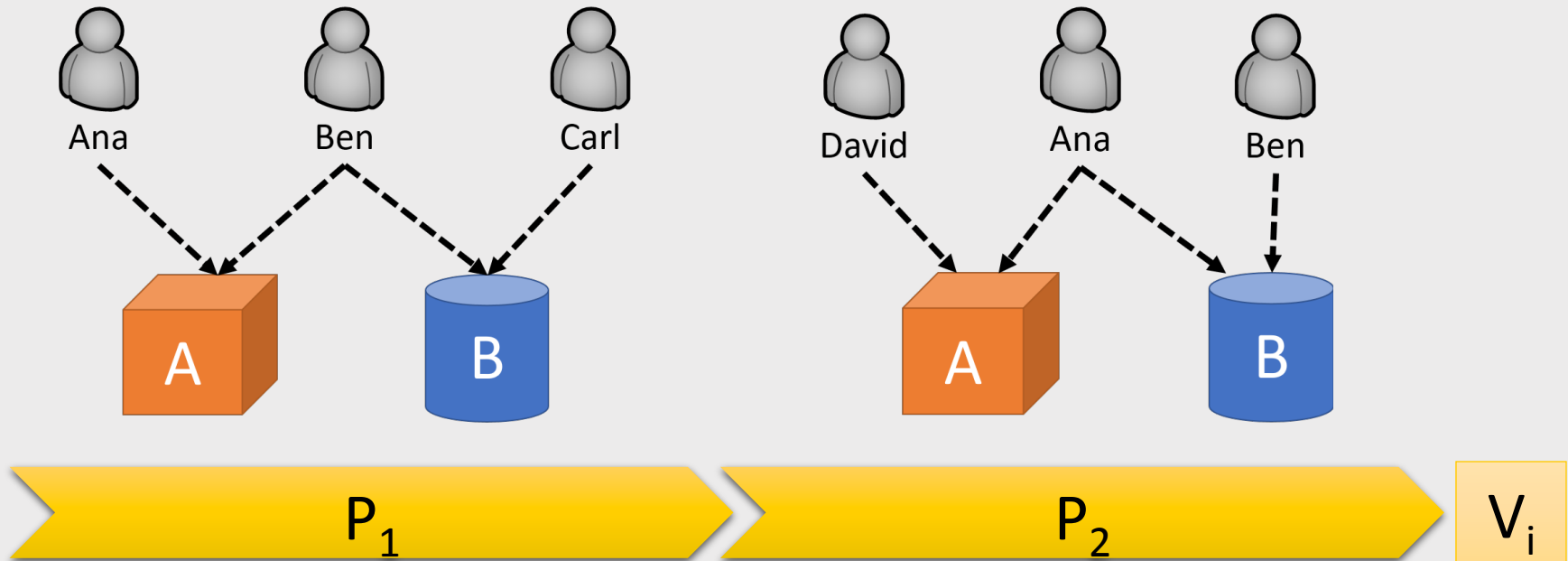
Problem 2: Define the Modules



Problem 2: Define the Modules

- Automatic decomposition based on co-change is ineffective if not enough files
- Solution: Manual decomposition
 - Modules should be file or directories (with or without sub-directories)
 - Use 3 judges: 3 independent decompositions
 - Merge the results

Turnover Metrics – Computation



1 External Newcomer (D)
1 Internal Leaver (B)

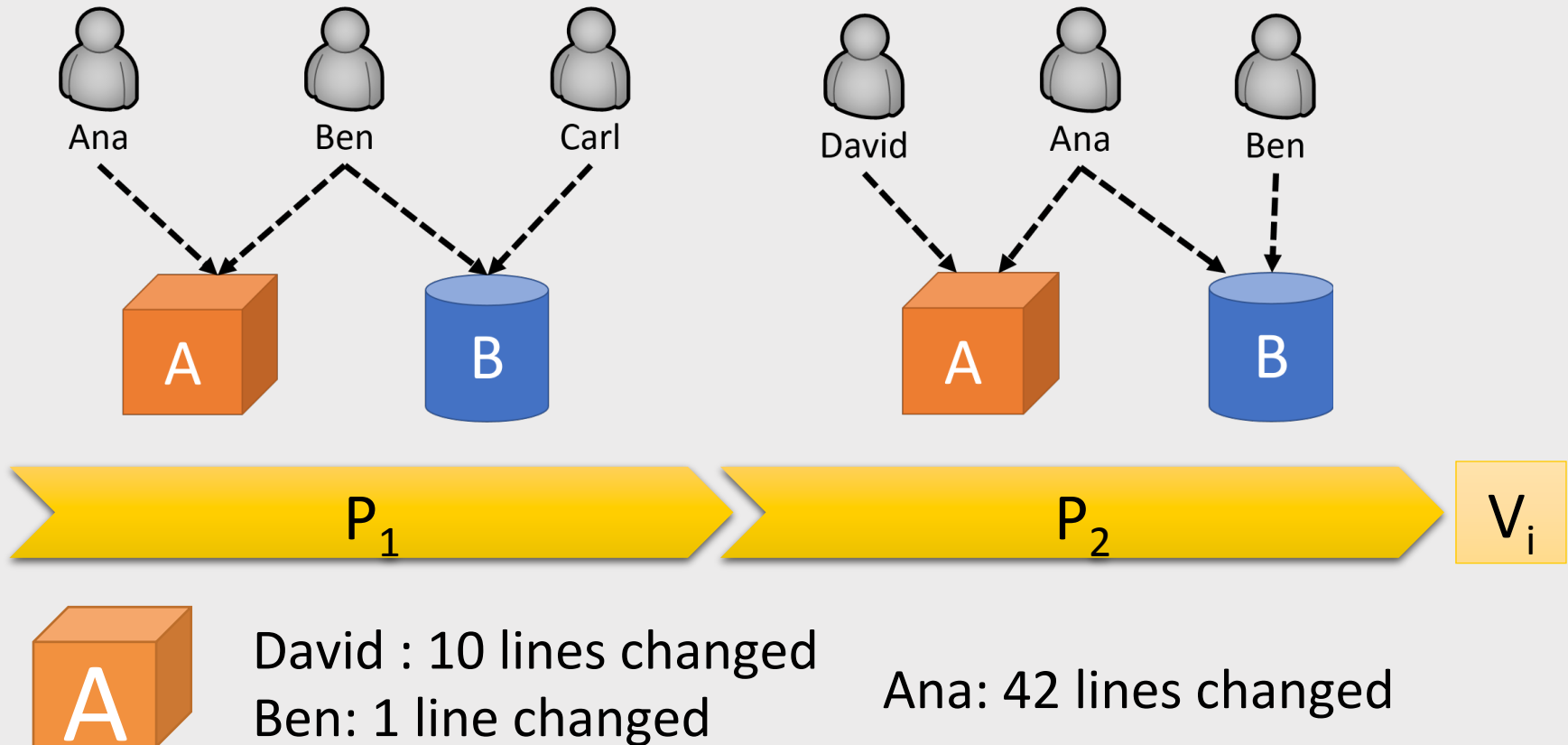
1 Stayer (A)



1 External Leaver (C)
1 Internal Newcomer (A)

1 Stayer
(B)

Turnover Metrics – Computation



Developers' activity is more accurate

Turnover Metrics – Formulae

$$ILA_{m,P_1,P_2} = \sum_{d \in IL_{m,P_1,P_2}} A_{m,d,P_1}$$

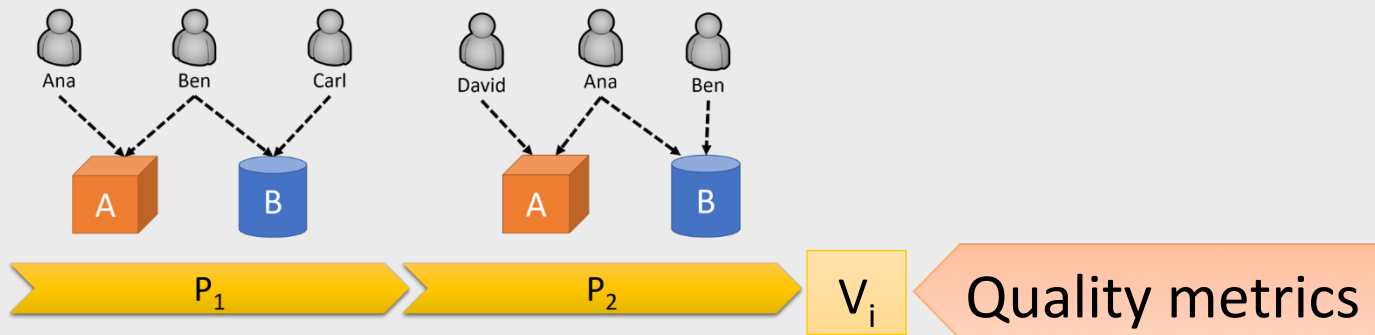
$$ELA_{m,P_1,P_2} = \sum_{d \in EL_{m,P_1,P_2}} A_{m,d,P_1}$$

$$INA_{m,P_1,P_2} = \sum_{d \in IN_{m,P_1,P_2}} A_{m,d,P_2}$$

$$ENA_{m,P_1,P_2} = \sum_{d \in EN_{m,P_1,P_2}} A_{m,d,P_2},$$

$$StA_{m,P_1,P_2} = \sum_{d \in St_{m,P_1,P_2}} avg(A_{m,d,P_1}, A_{m,d,P_2})$$

Methodology



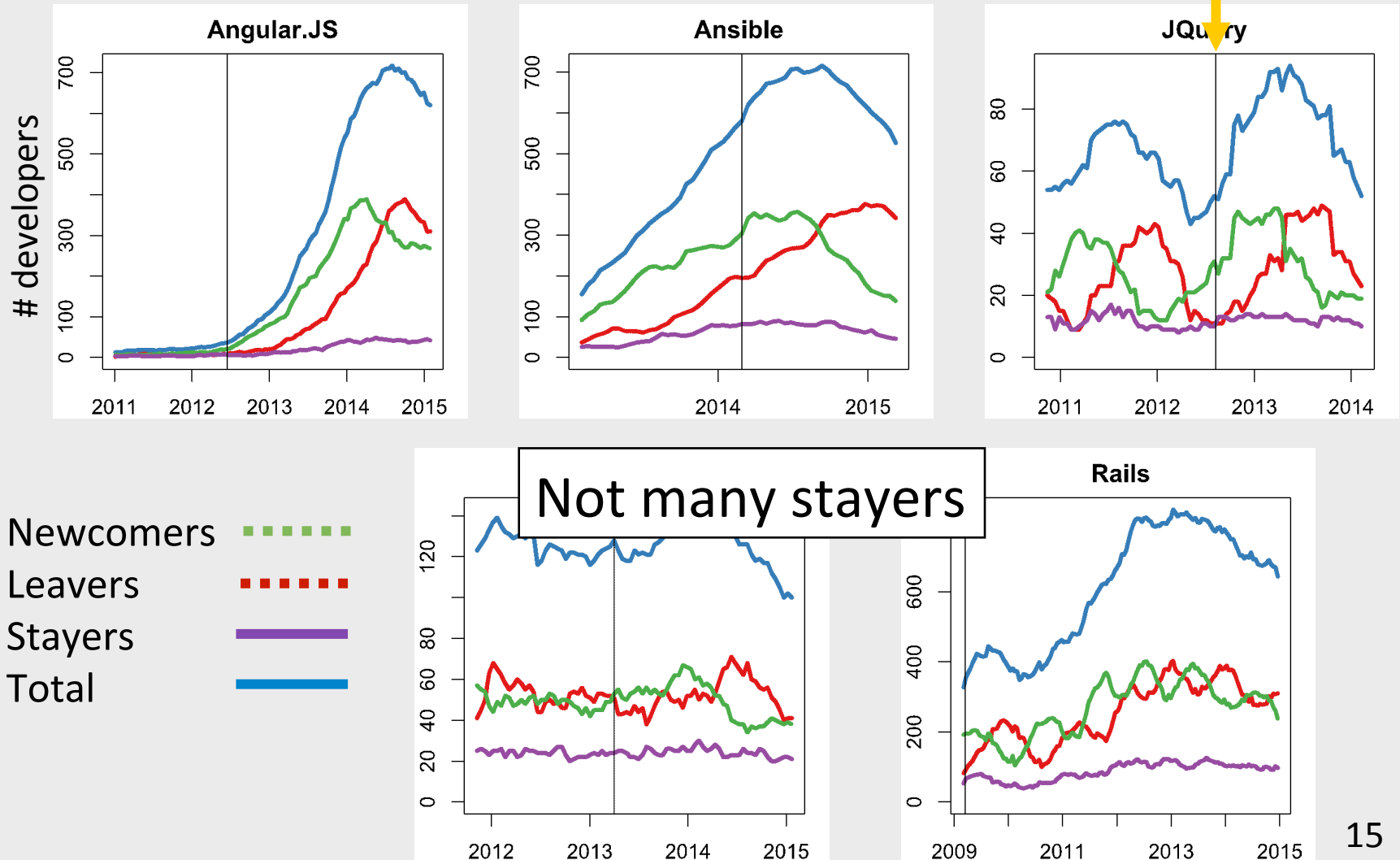
Module	ENA	INA	ELA	ILA	StA
A	12	34	21	33	9
B	25	0	1	12	40
C	7	222	21	123	0
D	20	9	0	8	90

Patterns of Turnover

Patterns of Turnover

- At the project level
 - For any version, at least 80% of contributors are either newcomers or leavers
 - Only 8% to 19% of newcomers become stayers at one point
 - The top stayers are mostly paid developers
- At the module level
 - Stayers are generally more active than newcomers or leavers (exception: Ansible)
 - All modules have either newcomers or leavers (or both)
 - External newcomers do not contribute alone

At the Project Level



At the Module Level

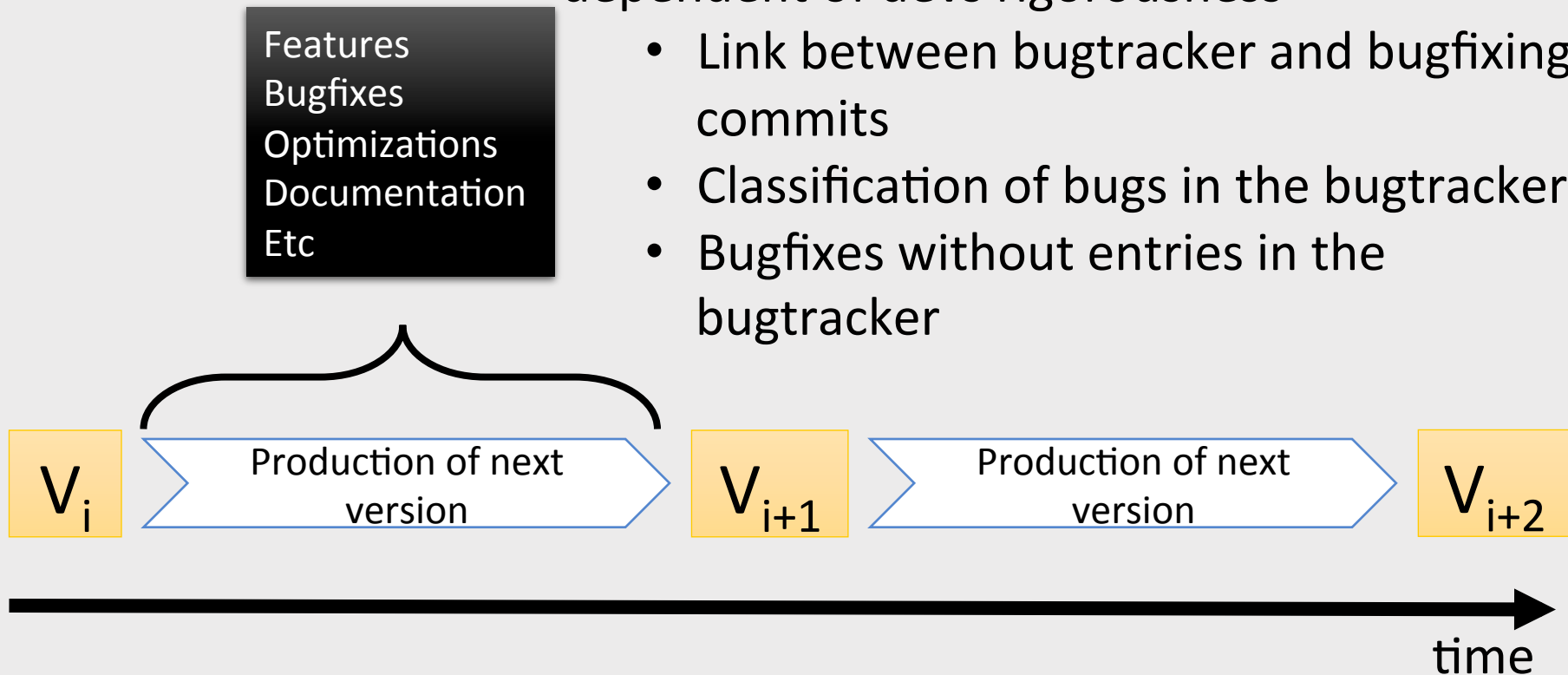
- Stayers are the most active
- External newcomers and leavers are focused on a subset of modules

Developer Turnover and Quality

Finding Bug-Fixing Commits

Automatic extraction is not reliable:
dependent of devs rigorousness

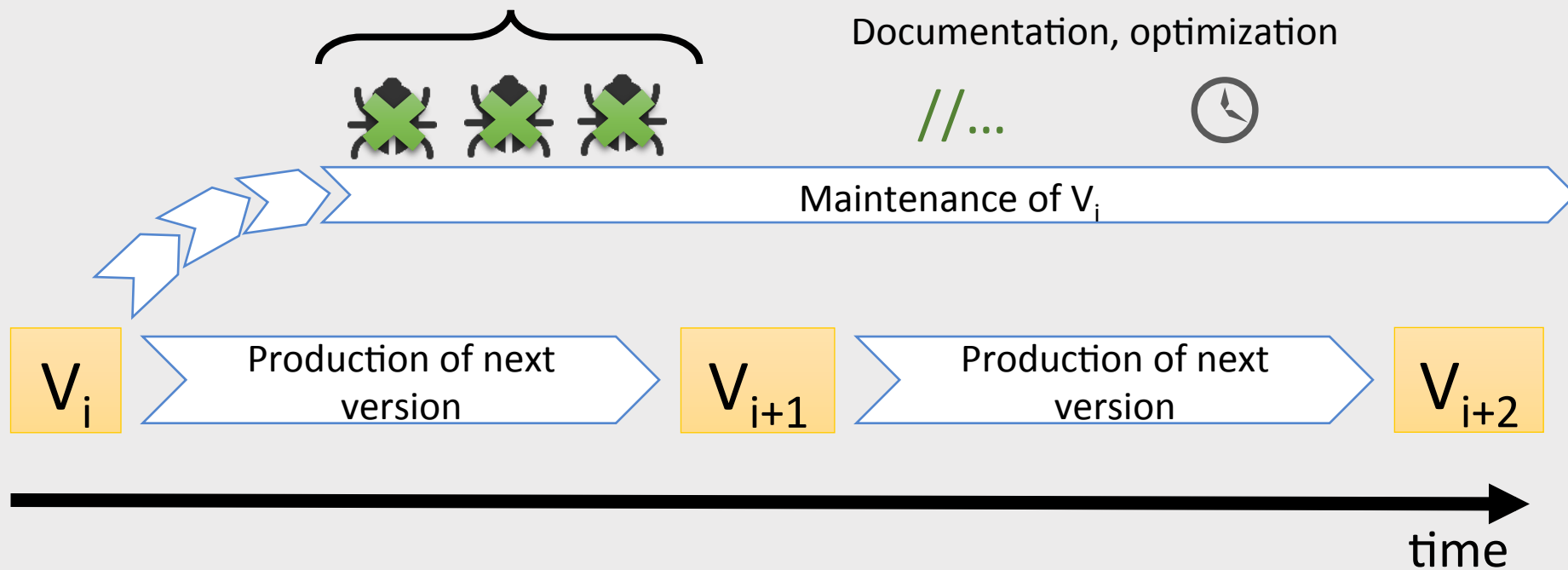
- Link between bugtracker and bugfixing commits
- Classification of bugs in the bugtracker
- Bugfixes without entries in the bugtracker



Finding Bug-Fixing Commits

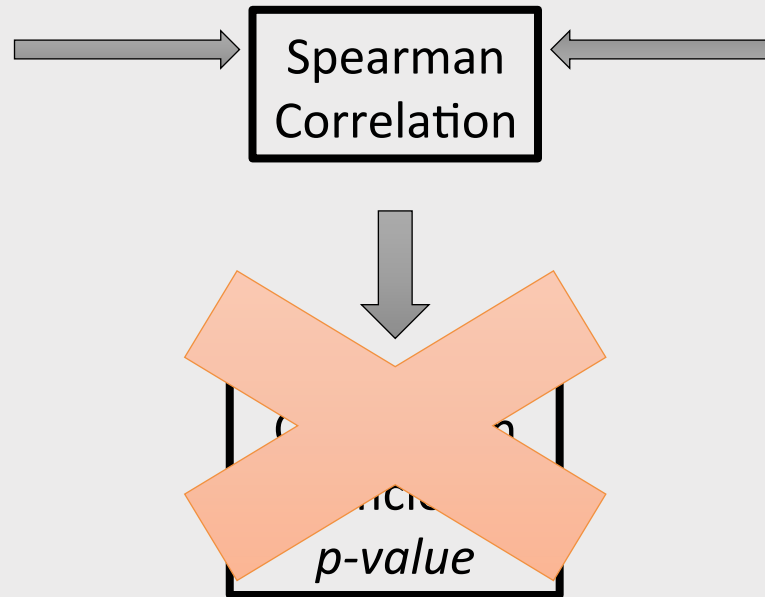
Manual Extraction

Bug-fixing commits



Relationship between Turnover and Quality

Module	ENA
A	12
B	25
C	7
D	20



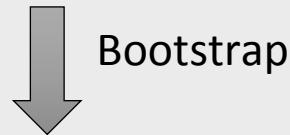
Module	Bug-fixing commit density
A	0.2
B	0.11
C	0.01
D	0.25

Relationship between Turnover and Quality

Module	ENA
A	12
B	25
C	7
D	20

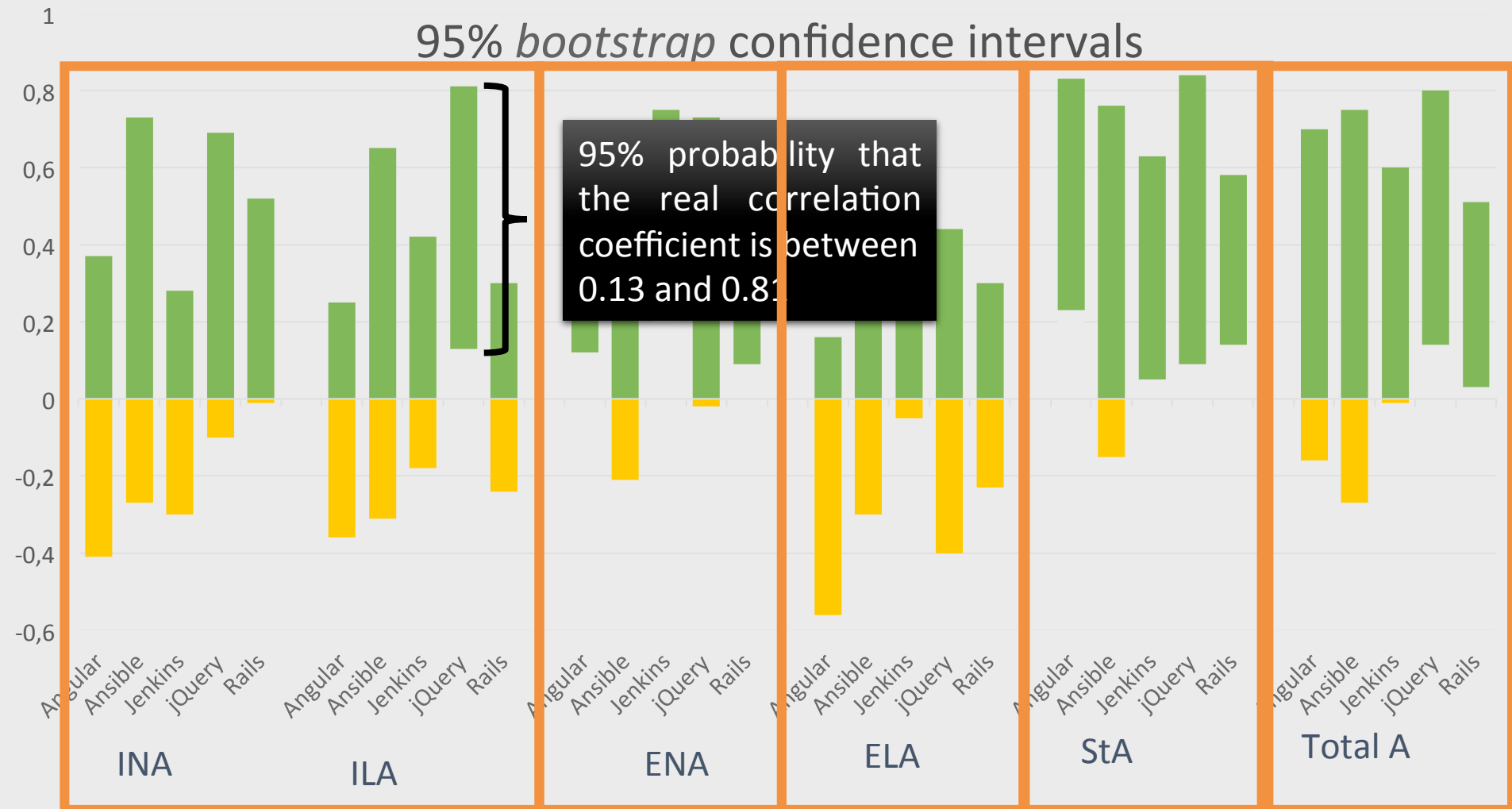


Module	Bug-fixing commit density
A	0.2
B	0.11
C	0.01
D	0.25



Confidence interval
for the correlation
coefficient

Correlation Between Turnover Metrics and Bug-fixing Commits Density



Study Conclusions

- A lot of newcomers and leavers, but not very active
- External newcomers have a strong relationship with quality
- External leavers did not show a significant relationship
- Internal turnover did not show a significant relationship

Future Work

- Identify different categories of developers
 - Paid and Volunteer
 - Core and occasional contributors
- Non-monotonic relationship
- Bug prediction

Replication Package

- Manually extracted data
 - Bug-fixing commits ids
 - Author identity merging information
 - Modules decomposition
- Extraction and analysis scripts

Data Extraction & Analysis

1. Browse commits from Git repository
2. Extract authors and code churn from commits
3. Compute metrics
4. Perform statistical analysis (correlation + bootstrap)

Extraction & Analysis Tool: *digg*

- Clones Git repositories
- Launches analyses for each repository
- Launches global analyses to produce final results
- Installation
 - `gem install diggit` 
 - <https://github.com/jrfaller/diggit.git>


Diggit Analyses

```
class AnalysisWithAddon < Diggit::Analysis
  require_addons 'db'

  def run
    db.do_something
    puts @options
    puts @source
    puts 'Runned!'
  end
end
```

@repo : Rugged Git repository (libgit2)
<http://www.rubydoc.info/gems/rugged>

Our Diggit Analyses

- Extracts developers activity
 - With modules lists
 - With authors identity merging
 - Extracts bugfixes
 - Extracts modules lines of code
- 
- mongoDB
- Global analysis: R script
 - Computes turnover metrics
 - Computes correlations
 - Produces figures

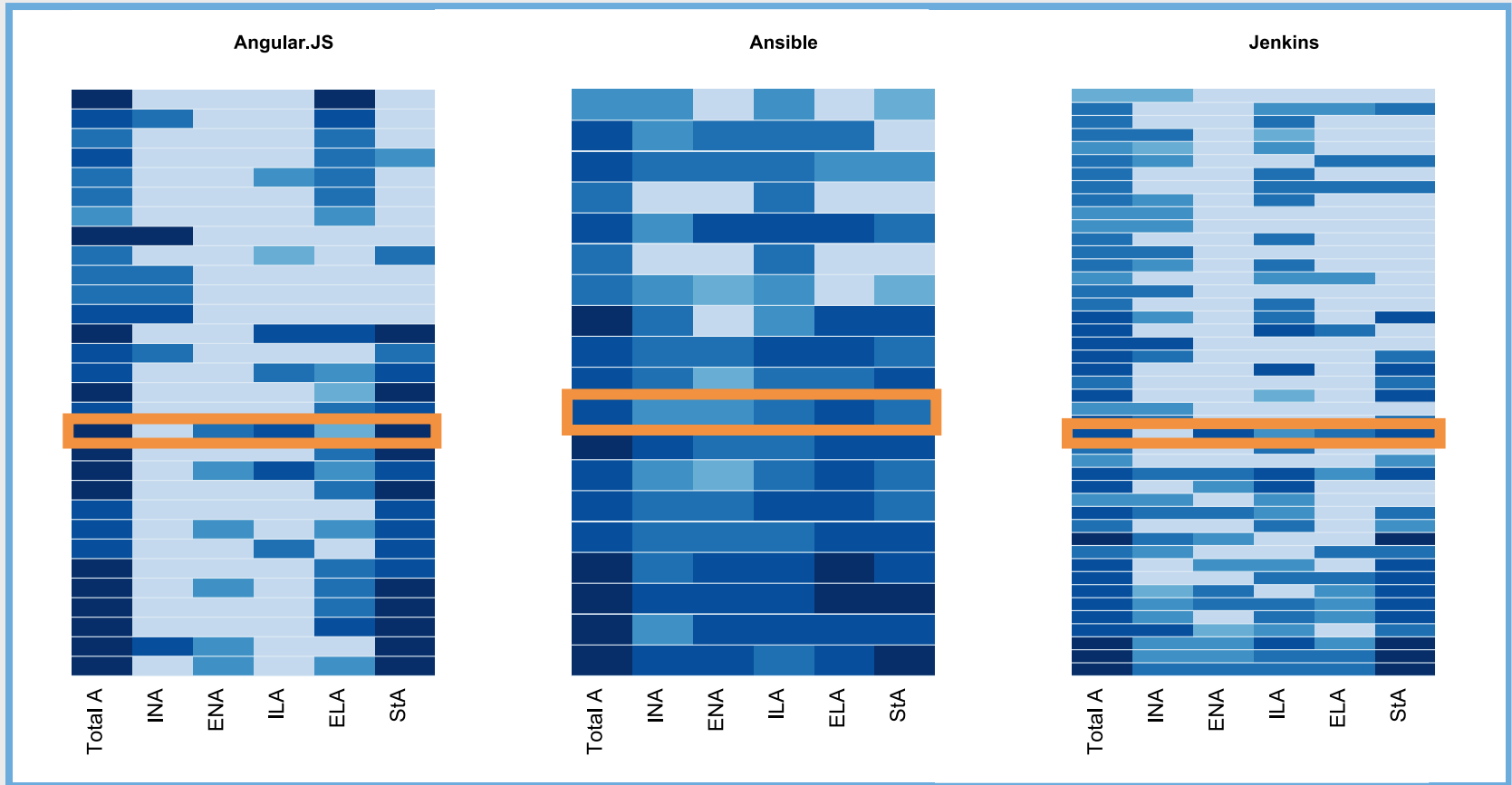
Replication Package

- VM image : <http://se.labri.fr/a/FSE15-foucalt/>
- Installation with gem
 - `gem install diggit`
 - `gem install diggit_developers-activity`
- Send me an email!
 - matthieu.foucalt@labri.fr

Talk Summary

- Turnover theories
- Measure turnover
 - Choose period size
 - Define modules
 - Extract developers activity level
- Turnover patterns
- Relationship with quality
 - Find bug-fixing commits
 - Estimate correlation
- Replicate

Patterns of Turnover



Patterns of Turnover

