



Software Integration Questions: A Quantitative Survey

Martín Dias, Verónica Uquillas-Gomez, Damien Cassou, Stéphane Ducasse

► **To cite this version:**

Martín Dias, Verónica Uquillas-Gomez, Damien Cassou, Stéphane Ducasse. Software Integration Questions: A Quantitative Survey. [Technical Report] Inria Lille. 2014. <hal-01093496>

HAL Id: hal-01093496

<https://hal.inria.fr/hal-01093496>

Submitted on 8 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Software Integration Questions: A Quantitative Survey

Martin Dias Verónica Isabel Uquillas Gómez
Damien Cassou Stéphane Ducasse

December 10, 2014

1 Introduction

Software is in constant evolution. In a software project, code changes represent bug fixes, enhancements, new features and adaptations due to changing domains. The evolution of a project codebase is usually managed in a *revision control system* that supports *branches*. Developers perform code changes in a branch and sometimes such changes are merged into other branch. This activity is called *integration*.

Integration of changes poses substantial challenges. We conducted a survey to evaluate a catalogue of 46 questions about integration. For each question, the participants had to rank the importance and the support that current tools offer.

In a period of 5 months we received the responses of 42 developers who integrate changes on very diverse software projects.

2 The Calls for Participation

We called for participation in several software development communities, which includes *Smalltalk-related* mailing-lists, *The Apache Software Foundation* twitter and *The Eclipse Foundation* twitter. Figure 1 and Figure 2 are examples of such calls for participation.

Date	April 22, 2013
Subject	Software Integration Survey
Body	<p>Hi all,</p> <p>We are conducting an academic research to improve tools for software integration. If you maintain a project with multiple contributors, we will greatly appreciate your participation.</p> <p>Your answers will be used for research purposes and to support the development of new tools. You can later access the results of this study, make free use of our tools to support the integration of changes, and maybe receive a gift.</p> <p>https://sondages.inria.fr/index.php/667625/lang-en</p> <p>Best regards, Martin Dias, Nicolas Anquetil, Veronica Isabel Uquillas, Stephane Ducasse and the RMOD Inria team</p>

Figure 1: First Call

Date	July 5, 2013
Subject	We need you for a software integration survey (reminder)
Body	<p>Hi all,</p> <p>Some time ago we sent an email to ask you contributing to a research to improve tools for software integration.</p> <p>If you maintain a project with multiple contributors, we will greatly appreciate your participation.</p> <p>Your answers will be used for research purposes and to support the development of new tools. You can later access the results of this study, make free use of our tools to support the integration of changes, and maybe receive a gift.</p> <p>We found out that the survey was not properly working on Safari (and may be also some other browsers). And we have now corrected this issue.</p> <p>Up to now, we received 36 full answers and 166 incomplete answers. So we decided to give you an extra month, if you started to fill the survey, please consider resuming filling the survey. For this you only need to go back to the same site: https://sondages.inria.fr/index.php/667625/lang-en</p> <p>If you did not yet fill in the survey, please consider doing it now.</p> <p>Best regards, Martin Dias, Nicolas Anquetil, Veronica Isabel Uquillas, Stephane Ducasse and the RMOD Inria team</p>

Figure 2: Second Call

3 The Survey

The Survey has two main parts: the "Integration Questions" and the "Participant Profile". Below, we briefly introduce such parts, and we include the complete survey that

the participants had to fill.

3.1 Integration Questions

This is the core part of the Survey. It includes 46 questions related which the participants evaluated in two dimensions:

Ranking	Abbreviation	Possible Answers
What is the importance of this question?	Importance	Nothing, Little, Moderate, Extreme.
Do your tools answer this question?	Tool Support	No, Partially, Yes.

Figure 3: Ranking for each Integration Question

The Integration Questions are split in the following groups:

- Changes Within a Stream
- Bug Tracking Infrastructure
- Change Nature
- Structural Change Characterization
- Authorship/Ownership

3.2 Participant Profile

In this part, we survey the profile of the participants. The Participant Profile Questions are split in the following groups:

- Integration of Changes
- Personal Background

3.3 Survey

Following, we include the survey that the participants filled. It contains 13 pages.

SOFTWARE INTEGRATION SURVEY

The activity of merging and integrating changes lacks of comprehensive support to assist developers. For example, the integration of changes can have an unexpected impact on the design or behavior of the system, leading to the introduction of bugs.

This survey is part of a joint academic research of the Vrije Universiteit Brussel (Belgium) and INRIA-Lille (France). We have already developed some open-source tools (MIT license) to ease the integration and merging activity [\[1\]](#), and we will continue working in the same direction.

We kindly ask you to respond this survey and greatly thank you in advance for your contribution. As an integrator, your participation is very important to improve our work. Your answers will be treated anonymously and confidentially. They will be used for research purposes and to support the development of other tools. You can later access the results of this study and make free use of our tools to support the integration of changes. To thank you for your contribution to this research, on August one participant to this survey will be randomly chose to be awarded with a gift check for Amazon.

If you have any question or concern do not hesitate to write at lse-survey@lists.gforge.inria.fr.

[\[1\] http://soft.vub.ac.be/~vuquilla/Site/Tools.html](http://soft.vub.ac.be/~vuquilla/Site/Tools.html)

There are 29 questions in this survey

Authorship/Ownership

These questions are related to the owner of the original code, and author of the commit.

[]

Please rank each question below.

(A1): The word "importance" refers to the support to the integration task that the answer of that question provides.

(A2): Indicates the coverage of your tools for answering the question.

Please choose the appropriate response for each item:

	(A1) What is the importance of this question?				(A2) Do your tools answer this question?		
	Nothing	Little	Moderate	Extreme	No	Partially	Yes
"Who is the author of this changed code?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Who was the previous owner of the changed code?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Has my own code been changed?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"What is the general quality of the change committer?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"How many people have contributed to this group of commits?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 4: Survey, page 1 of 13.

Structural change characterization

These questions relate to the structure of the original code as well as the changes. They cover various aspects in terms of volume, impact, dependencies (which packages, classes should be loaded before), and so on. These questions relate to the structure of the original code as well as the changes. They cover various aspects in terms of volume, impact, dependencies (which packages, classes should be loaded before), and so on.

Note: A **commit** is a group of changes.

[]

Please rank each question below.

(A1): The word "importance" refers to the support to the integration task that the answer of that question provides.

(A2): Indicates the coverage of your tools for answering the question.

Please choose the appropriate response for each item:

	(A1) What is the importance of this question?				(A2) Do your tools answer this question?		
	Nothing	Little	Moderate	Extreme	No	Partially	Yes
"How large is the commit?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"How many entities (packages/classes/methods) are impacted by the commit? (Impacted in the sense they can they stop compiling, for example)"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Is this commit confined to a single package or spread over the entire system?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"What is the complexity of the changes?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Do all the changes within the commit belong together? (Can we split the commit?)"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Are there other packages that I will need to change as well to integrate this commit? (Can we identify the users of the changed code?)"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Will the code compile after applying this commit?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Is the commit conflict free? (Does this change generate any syntactic merge conflicts when integrating?)"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Which entities (packages/classes/methods) have been changed?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Does this change depend on other changes (in the source branch) to be functional (in the target branch)?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 5: Survey, page 2 of 13.

Change nature

These questions relate to the nature, behavior and intent of a change. Note that some of these questions are open-ended and therefore inherently difficult to answer automatically.

[]

Please rank each question below.

(A1): The word "importance" refers to the support to the integration task that the answer of that question provides.

(A2): Indicates the coverage of your tools for answering the question.

Please choose the appropriate response for each item:

	(A1) What is the importance of this question?				(A2) Do your tools answer this question?		
	Nothing	Little	Moderate	Extreme	No	Partially	Yes
"Does the commit follow rules and conventions?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Is the vocabulary used in the commit consistent with the one of the system?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Does this commit improve the quality of the system?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Does this commit correctly fulfill its goal? (Does it fix correctly a particular problem?)"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"What is the intention of this commit?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"In a commit with 'strange code', was the strange code intentional (it has to be like that to turn around a special aspect of the system), or accidental (the author did not really know what he was doing)?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"What kind of commit is it? (Bug fix/New feature/Refactoring/Documentation/...)"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Does this commit fix/break tests? Which tests?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Is the commit covered by tests? What is the coverage? How can I test it?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"If I apply the commit, what are the parts of my current system that it affect? What are the users (classes/methods/functions) potentially impacted by this change in the destination branch/fork?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"What are the implications of this commit on the (potentially undeclared) API? (Are there any unknown users of the API that will be impacted by the changes?)"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 6: Survey, page 3 of 13.

Bug tracking infrastructure

These questions are related to bug tracking facilities.

[]

Please rank each question below.

(A1): The word "importance" refers to the support to the integration task that the answer of that question provides.

(A2): Indicates the coverage of your tools for answering the question.

Please choose the appropriate response for each item:

	(A1) What is the importance of this question?				(A2) Do your tools answer this question?		
	Nothing	Little	Moderate	Extreme	No	Partially	Yes
"To which bug entry does this change relate?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"What bug fixes also affected the part of the system that is being impacted by this change?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 7: Survey, page 4 of 13.

Changes within a stream

These questions situate the changes within the context of a stream of changes (set of subsequent commits), as well as to the time at which the change occurs. In particular when working on a stream of changes, these questions capture the place of a change within the stream.

[]

Please rank each question below.

(A1): The word "importance" refers to the support to the integration task that the answer of that question provides.

(A2): Indicates the coverage of your tools for answering the question.

Please choose the appropriate response for each item:

	(A1) What is the importance of this question?				(A2) Do your tools answer this question?		
	Nothing	Little	Moderate	Extreme	No	Partially	Yes
"How old is this commit (compared to the version to which it should be integrated)?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"In which commit/version of the system was this method/function previously changed?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Did this class/method/function change (a lot) recently/in the past?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Is this change to a class/method/function the most recent one (in the branch)?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Is there any pending commit in the sequence of commits (in the branch) that might supersede this one?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Is this commit part of a whole series of commits?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Does this commit depend on previous ones? (What are the other commits needed first to merge this commit?)"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Is the change to a class/method/function ever used in subsequent changes?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Is this change to a class/method/function reverting the code to an old state?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"What else changed when this code was introduced or modified (i.e., documentation, website, database schema)?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"What other classes/methods/functions changed when this code was introduced or modified?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"What are the other changes made by the same author/during	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 8: Survey, page 5 of 13.

the same time period?"							
"Did the changing classes/methods/functions of this commit change together in a previous commit?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"If there were changes to classes/methods/functions happening together in the past, can we suspect that there is still something missing in the current commit?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"Were the classes/methods/functions affected by this change renamed in the past and if so, in which version of the system?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"What were the users (callers) of a changed method/function in a particular version of the system?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"What are the current users (callers) of a changed method/function?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
"What commits of another branch have been integrated into this branch?"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

[]Are there other question(s) that you ask when integrating changes and that we are missing in our catalogue?

Please write your answer here:

Figure 9: Survey, page 6 of 13.

Integration of Changes

Please answer the following questions about the software system in which you integrate changes.

[] In what kind of software do you integrate changes from other developers?

Please choose **all** that apply:

End user applications (Web, desktop, mobile, ...)

Libraries, frameworks, platforms

Embedded

Other:

[] Is the project an open-source system?

Please choose **only one** of the following:

Yes

No

[] What is the approximated size of the system?

Please write your answer(s) here:

LOCs

Classes

Files

Packages

[] How many developers are working on this system?

Please write your answer here:

Figure 10: Survey, page 7 of 13.

<p>[]How many integrators are working on this system?</p> <p>Please write your answer here:</p> <input type="text"/>									
<p>[]Are you currently one of the developers of this system?</p> <p>Please choose the appropriate response for each item:</p> <table><tr><td></td><td>Yes</td><td>No</td></tr><tr><td>if YES, are you the main developer?</td><td><input type="radio"/></td><td><input type="radio"/></td></tr><tr><td>if NO, did you work previously on the system?</td><td><input type="radio"/></td><td><input type="radio"/></td></tr></table>		Yes	No	if YES, are you the main developer?	<input type="radio"/>	<input type="radio"/>	if NO, did you work previously on the system?	<input type="radio"/>	<input type="radio"/>
	Yes	No							
if YES, are you the main developer?	<input type="radio"/>	<input type="radio"/>							
if NO, did you work previously on the system?	<input type="radio"/>	<input type="radio"/>							
<p>[]Are you the main integrator of this system? (Integrator by opposition to Developers)</p> <p>Please choose only one of the following:</p> <p><input type="radio"/> Yes</p> <p><input type="radio"/> No</p>									
<p>[]Which versioning tool(s) are using for this system?</p> <p>Please choose all that apply:</p> <p><input type="checkbox"/> AccuRev</p> <p><input type="checkbox"/> Bazaar</p> <p><input type="checkbox"/> ClearCase</p> <p><input type="checkbox"/> CVS</p> <p><input type="checkbox"/> darcs</p> <p><input type="checkbox"/> Git</p> <p><input type="checkbox"/> Mercurial</p> <p><input type="checkbox"/> Monticello</p> <p><input type="checkbox"/> Subversion</p> <p><input type="checkbox"/> Team Foundation Server</p> <p><input type="checkbox"/> Other: <input type="text"/></p>									

Figure 11: Survey, page 8 of 13.

[] How often does the integration happen?

Please choose **all** that apply:

- Periodically: Daily
- Periodically: Weekly
- Periodically: Monthly
- Ad-hoc: when needed or convenient
- Event-driven: on product milestones
- On-Completion: when the work in the branch is fully completed
- Other:

[] What is the nature of changes that are integrated?

Please choose **all** that apply:

- Bug fixes
- New features
- Feature enhancement
- Platform changes
- Cosmetic changes
- Other:

[] What types of branches are defined for this system?

Please choose **all** that apply:

- Development
- Feature (to develop each new feature in a specific branch)
- Bug fix
- Merge (to merge different branches)
- Release (to maintain specific versions of a product)
- Experimental/Prototype (to isolate disruptive or unstable code that may not go into production)
- Contributor/Personal (to isolate individual work before sharing it with the product team)
- Platform (to separate product configurations or contexts)
- Other:

Figure 12: Survey, page 9 of 13.

[]How do you merge changes?

Please choose **all** that apply:

- Downstream merges: pull changes into lower branch levels (e.g. from the mainline into a feature branch)
- Upstream merges: push changes into higher branch levels (e.g. from a feature branch into the mainline)
- Other:

[]Does the system have forks? If so, how many?

Please write your answer here:

A fork is a branch that has diverged from the original system. For example: Ubuntu from Debian, Firefox from Mozilla Application Suite, etc.

[]How often do you integrate/merge changes from a forked system?

Please choose **all** that apply:

- Periodically: Daily
- Periodically: Weekly
- Periodically: Monthly
- Ad-hoc: when needed or convenient
- Event-driven: on product milestones
- On-Completion: when the work in the branch is fully completed
- Never
- Other:

Figure 13: Survey, page 10 of 13.

[] Do you interact with developers when integrating/merging changes? if yes, when in particular (eg. to solve conflicts, to provide you with information about changes, ...)?

Please choose **only one** of the following:

- Yes
- No

Make a comment on your choice here:

[] What are the most significant problems that you had have with merges?

Please choose **all** that apply:

- MERGE CONFLICTS (occur when parallel changes to the same areas of code are incompatible)
- TEST REGRESSIONS (refer to failures in a test suite during or after a merge)
- CROSS-CUTTING REGRESSIONS (affect multiple parts of the product and are difficult to capture in a test suite, such as performance or security degradations)
- COMPILATION ERRORS (result from improper merge conflict resolutions that produce syntactically incorrect code, or from errors introduced into the build scripts themselves)
- Other:

Figure 14: Survey, page 11 of 13.

Personal Background

Please answer the following questions with regard to your background. The answers will be kept private and only serve to put your other answers in context.

[] e-mail (optional, only needed if you want to receive a summary of the results, or you are interested in being selected for the gift)

Please check the format of your answer.

Please write your answer here:

[] What is your age?

Please write your answer here:

[] What is your gender?

Please choose **only one** of the following:

- Male
- Female

[] What is your academic background?

Please choose **only one** of the following:

- No university degree
- Bachelor
- Master
- Engineer
- PhD
- Other

Figure 15: Survey, page 12 of 13.

[] How many years have you been developing software?

Please write your answer here:

[] How long have you been integrating changes?

Please write your answer(s) here:

Months

Years

[] In which programming languages do you consider an expert?

Please choose **all** that apply:

C

C++

C#

Clojure

Erlang

Haskell

Java

Lisp

ObjectiveC

PHP

Python

Ruby

Smalltalk

Other:

Figure 16: Survey, page 13 of 13.

4 The Responses

In this section, we summarise the responses of the 42 participants in this survey. We include three charts for each one of the 46 questions, from Figure 19 to Figure 64. To better understand the mentioned charts, Figure 17 and Figure 18 show the abbreviations used.

Importance	What is the importance of this question?
N	Nothing
L	Little
M	Moderate
E	Extreme

Figure 17: Legend for figures of "Importance"

Tool Support	Do your tools answer this question?
N	No
P	Partially
Y	Yes

Figure 18: Legend for figures of "Tool Support"

4.1 Authorship/Ownership

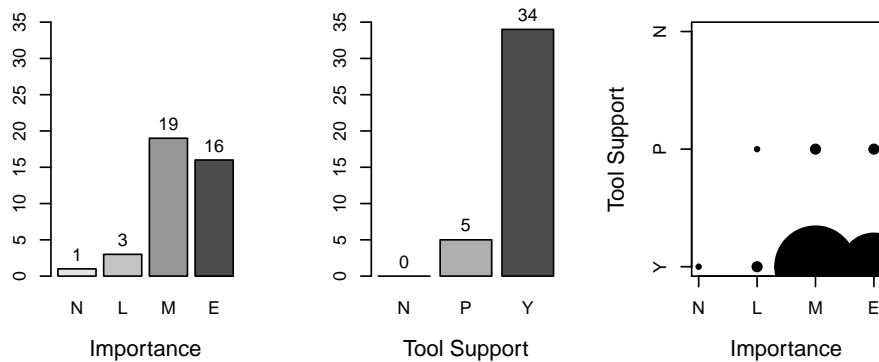


Figure 19: Who is the author of this changed code?

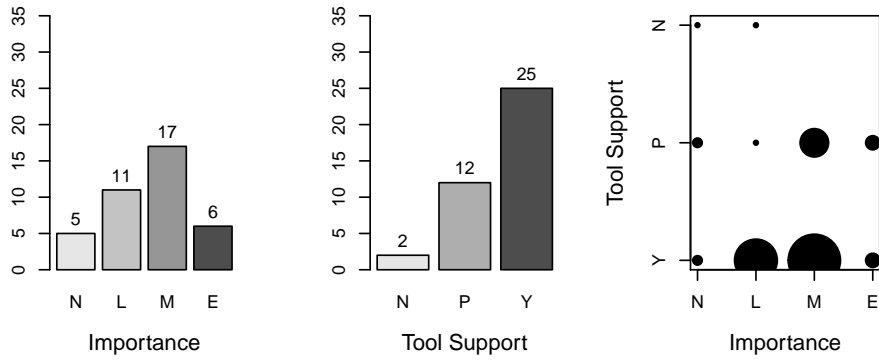


Figure 20: Who was the previous owner of the changed code?

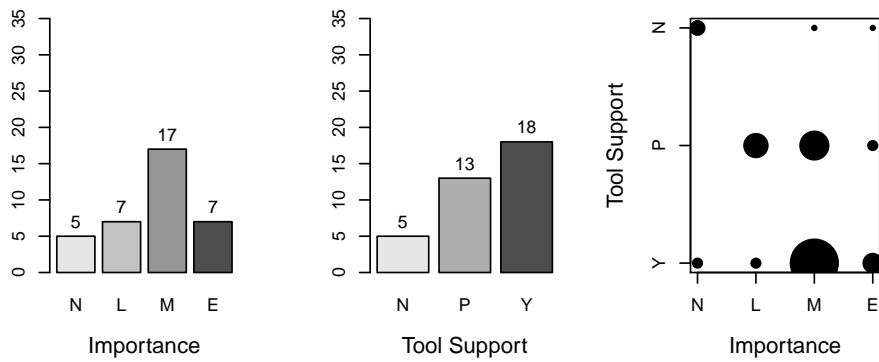


Figure 21: Has my own code been changed?

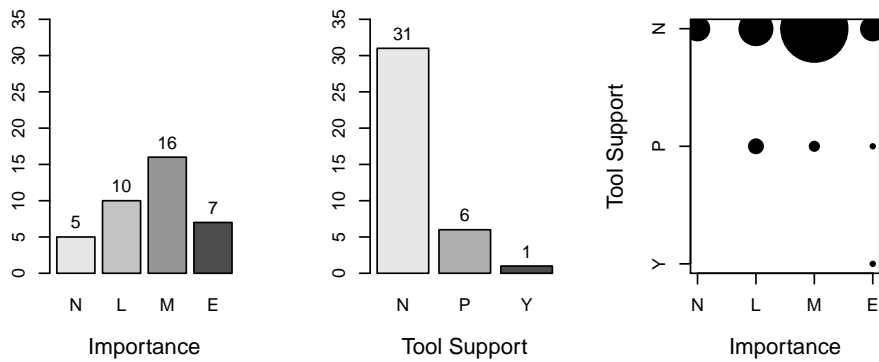


Figure 22: What is the general quality of the change committer?

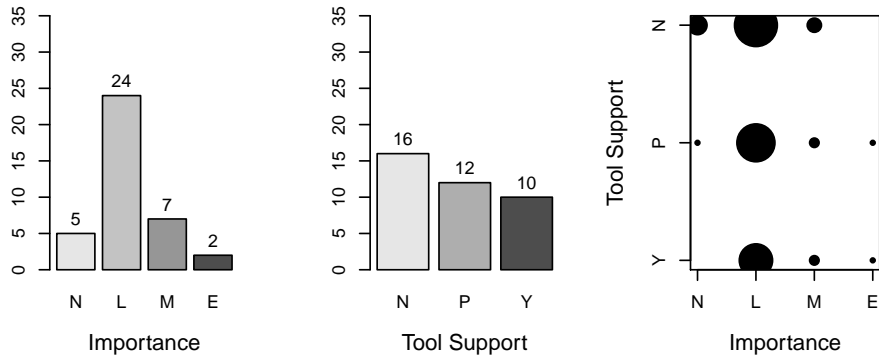


Figure 23: How many people have contributed to this group of commits?

4.2 Structural Change Characterization

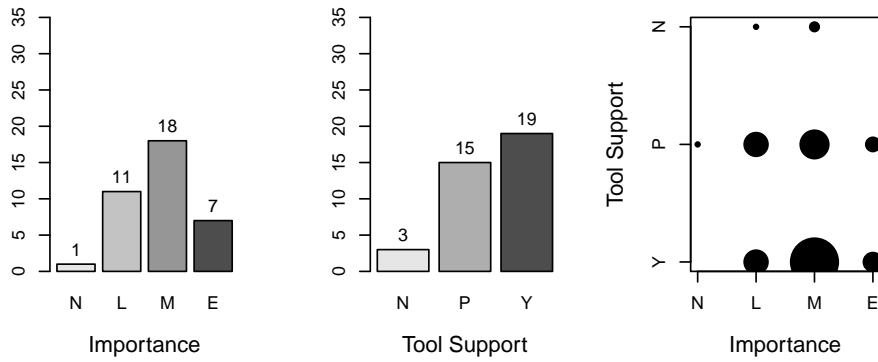


Figure 24: How large is the commit?

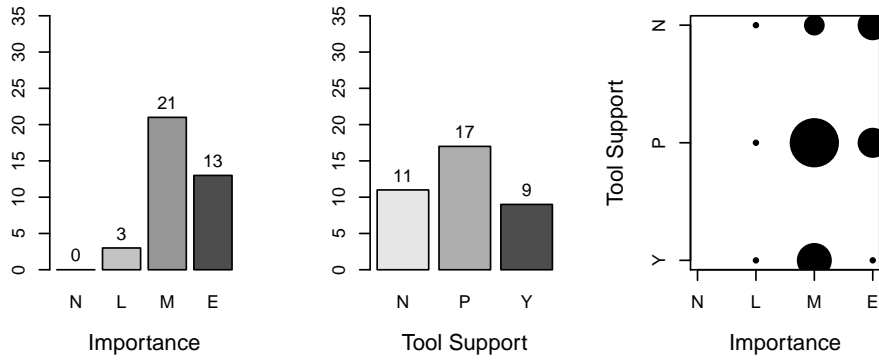


Figure 25: How many entities (packages/classes/methods) are impacted by the commit? (Impacted in the sense they can they stop compiling, for example)

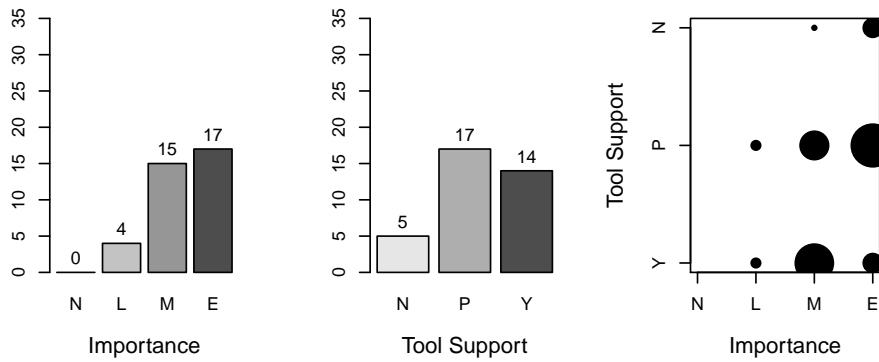


Figure 26: Is this commit confined to a single package or spread over the entire system?

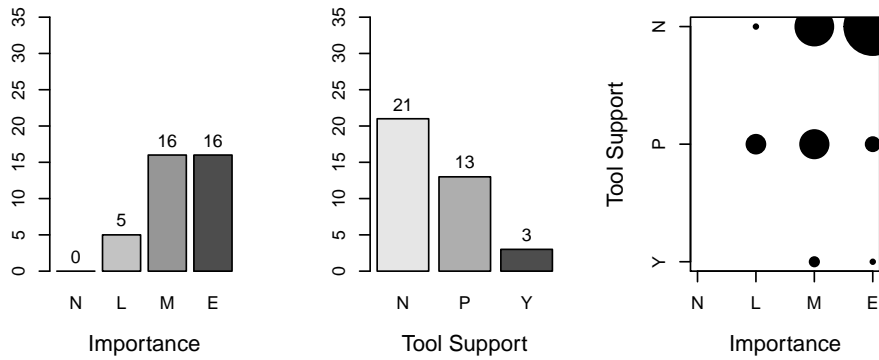


Figure 27: What is the complexity of the changes?

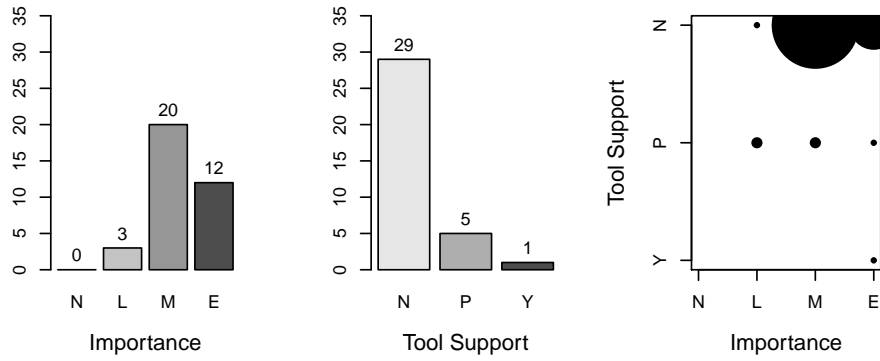


Figure 28: Do all the changes within the commit belong together? (Can we split the commit?)

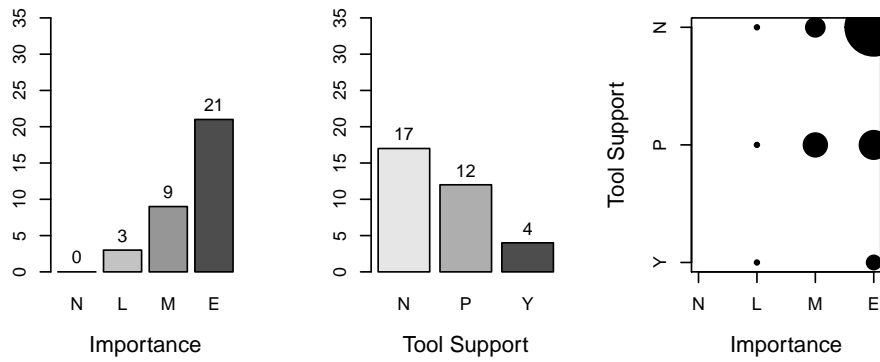


Figure 29: Are there other packages that I will need to change as well to integrate this commit? (Can we identify the users of the changed code?)

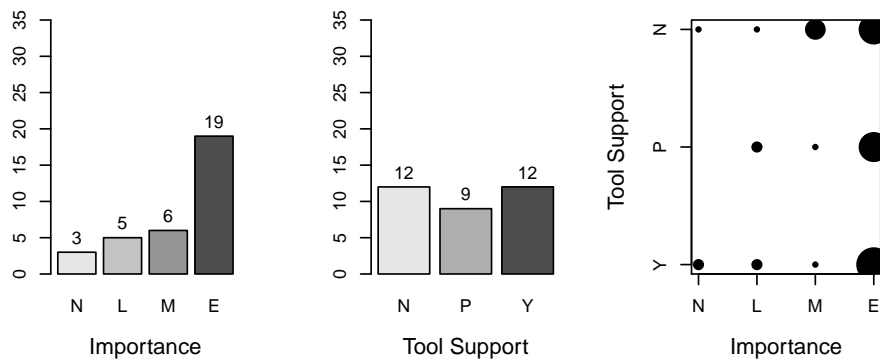


Figure 30: Will the code compile after applying this commit?

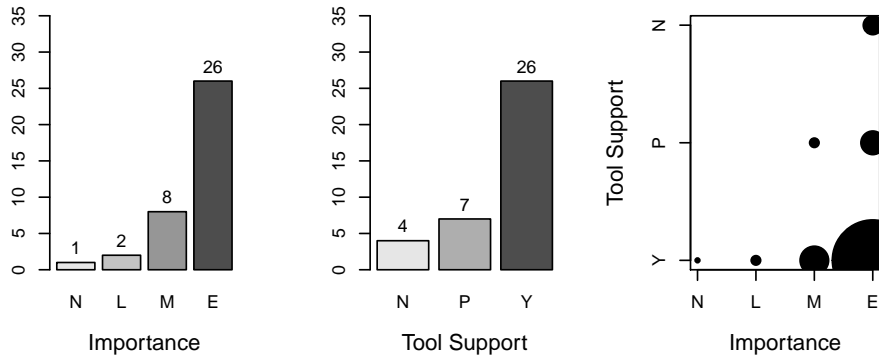


Figure 31: Is the commit conflict free? (Does this change generate any syntactic merge conflicts when integrating?)

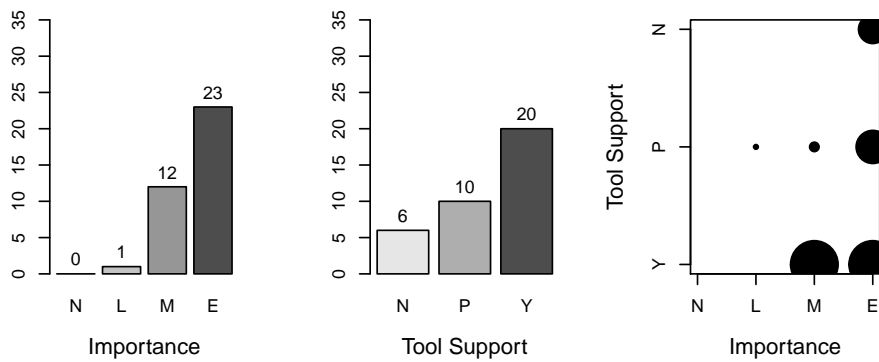


Figure 32: Which entities (packages/classes/methods) have been changed?

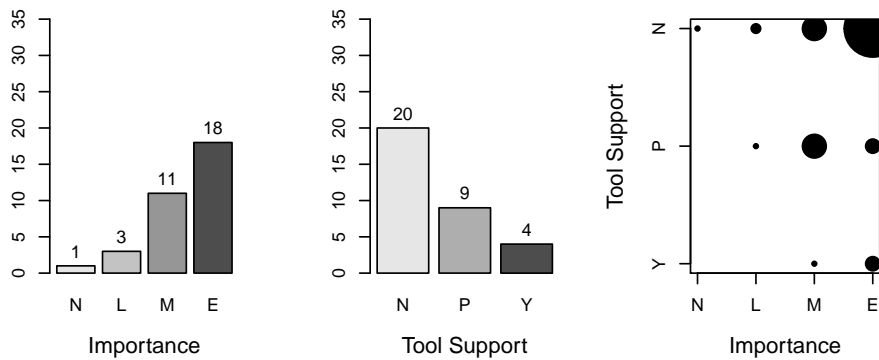


Figure 33: Does this change depend on other changes (in the source branch) to be functional (in the target branch)?

4.3 Change Nature

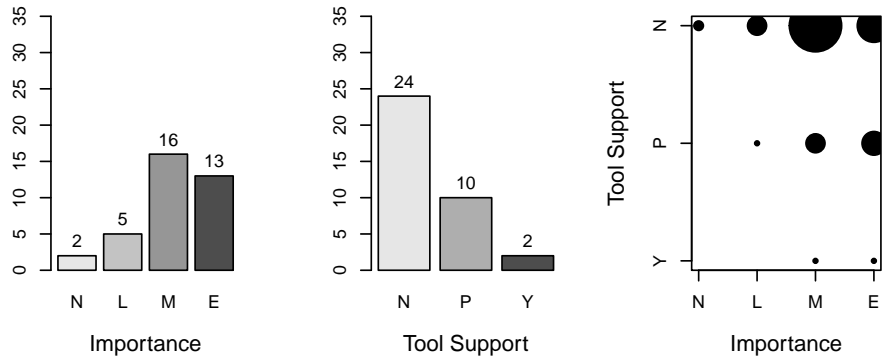


Figure 34: Does the commit follow rules and conventions?

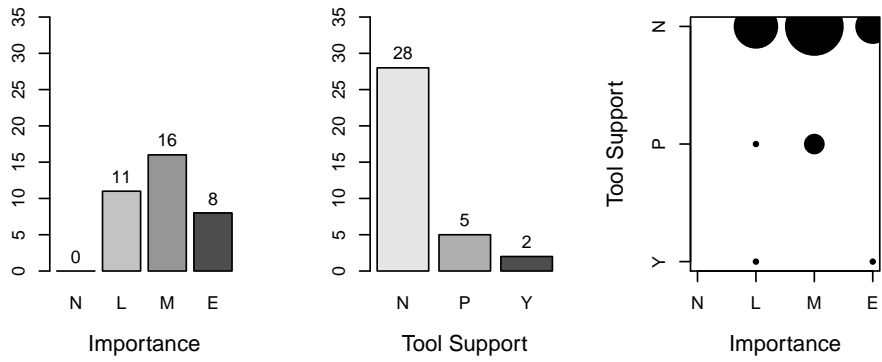


Figure 35: Is the vocabulary used in the commit consistent with the one of the system?

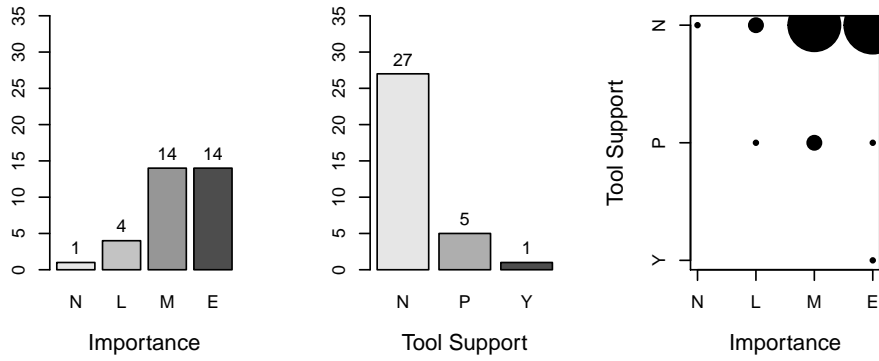


Figure 36: Does this commit improve the quality of the system?

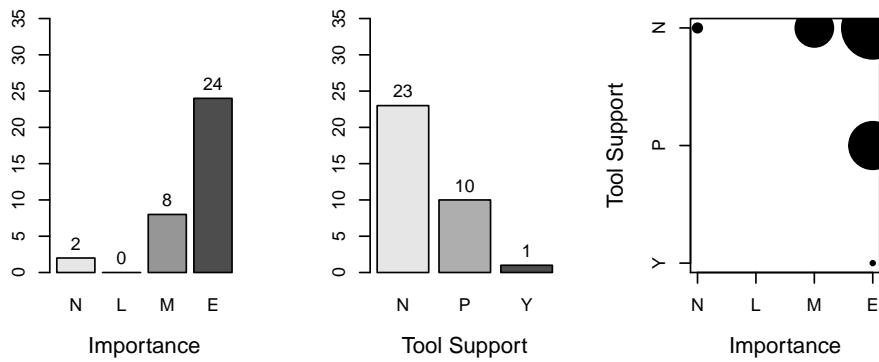


Figure 37: Does this commit correctly fulfill its goal?(Does it fix correctly a particular problem?)

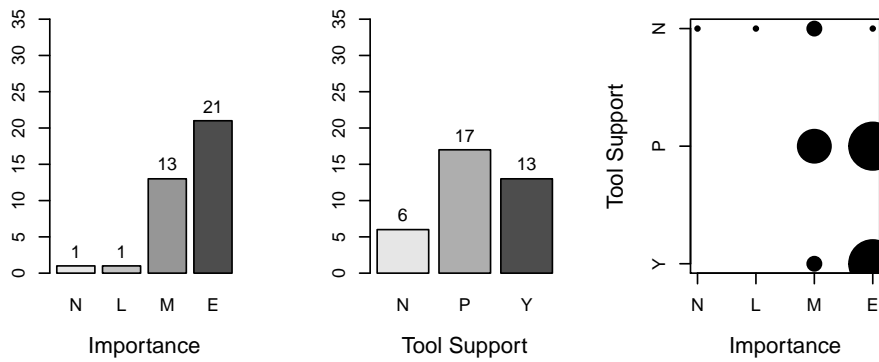


Figure 38: What is the intention of this commit?

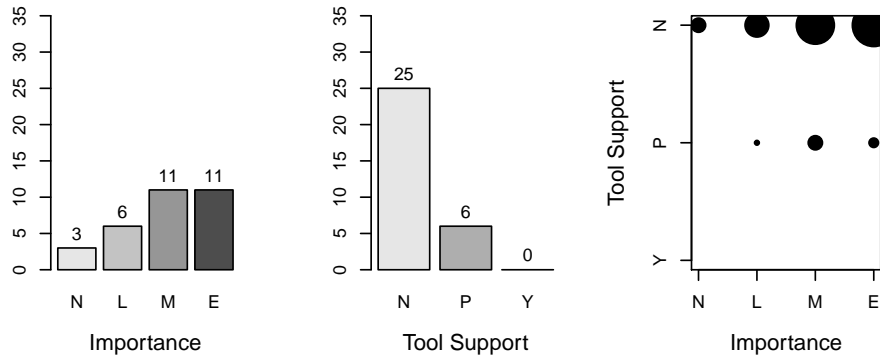


Figure 39: In a commit with strange code, was the strange code intentional (it has to be like that to turn around a special aspect of the system), or accidental (the author did not really know what he was doing)?

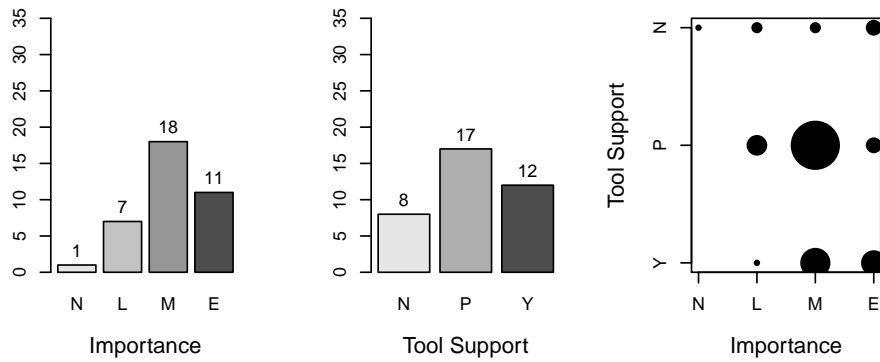


Figure 40: What kind of commit is it? (Bug fix/New feature/Refactoring/Documentation/...)

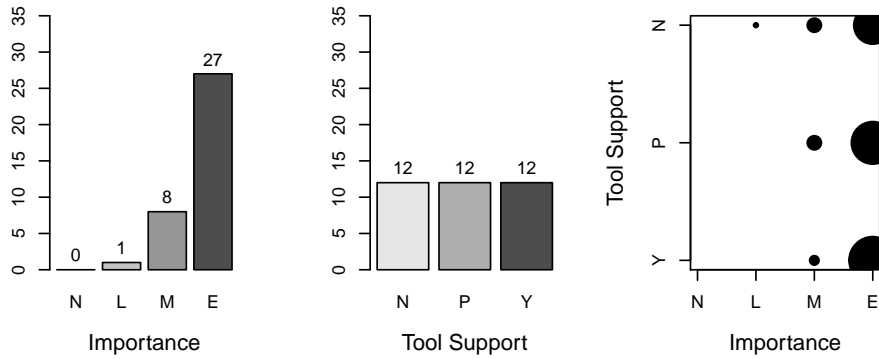


Figure 41: Does this commit fix/break tests? Which tests?

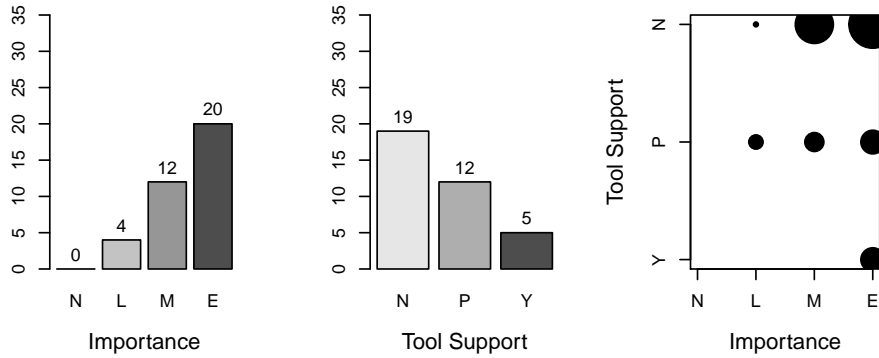


Figure 42: Is the commit covered by tests? What is the coverage? How can I test it?

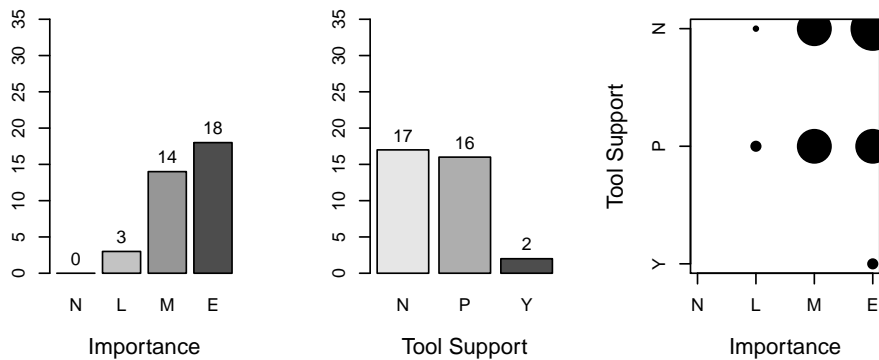


Figure 43: If I apply the commit, what are the parts of my current system that it affect? What are the users (classes/methods/functions) potentially impacted by this change in the destination branch/fork?

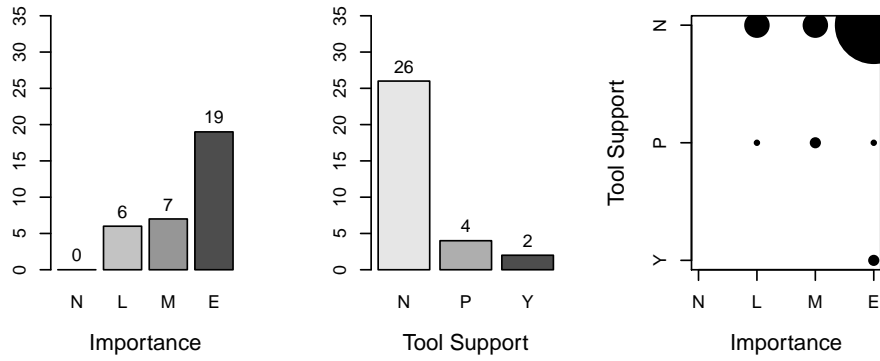


Figure 44: What are the implications of this commit on the (potentially undeclared) API? (Are there any unknown users of the API that will be impacted by the changes?)

4.4 Bug Tracking Infrastructure

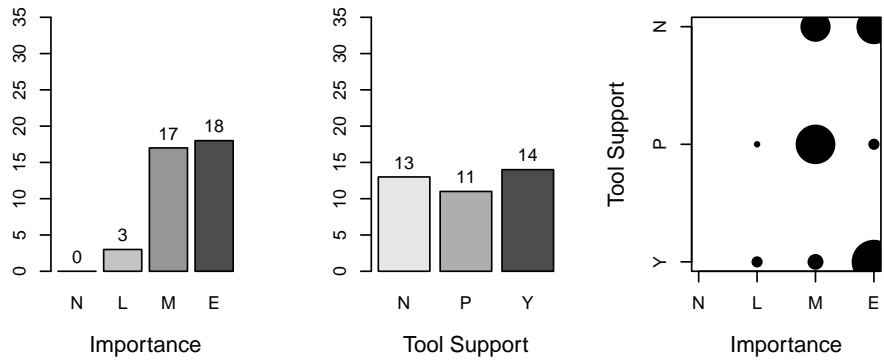


Figure 45: To which bug entry does this change relate?

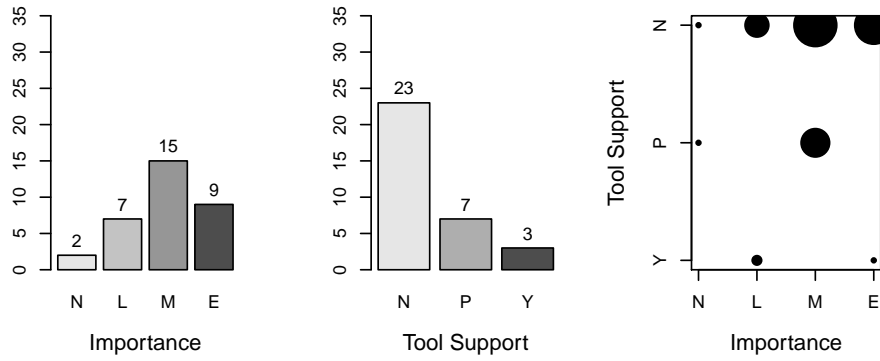


Figure 46: What bug fixes also affected the part of the system that is being impacted by this change?

4.5 Changes Within a Stream

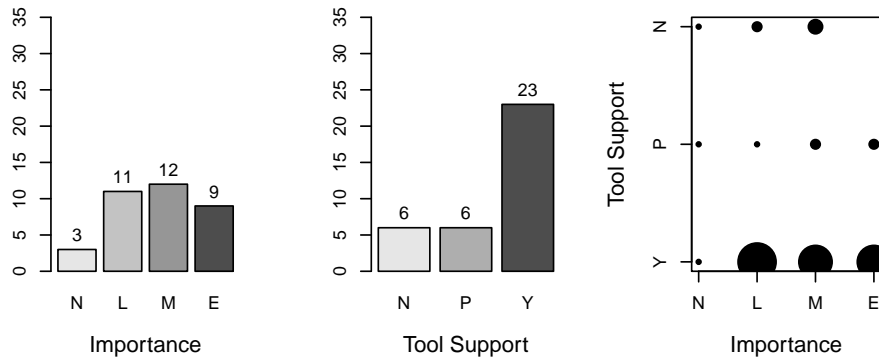


Figure 47: How old is this commit (compared to the version to which it should be integrated)?

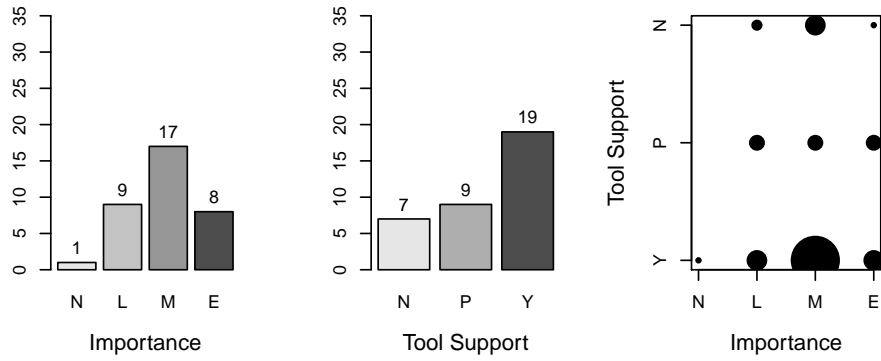


Figure 48: In which commit/version of the system was this method/function previously changed?

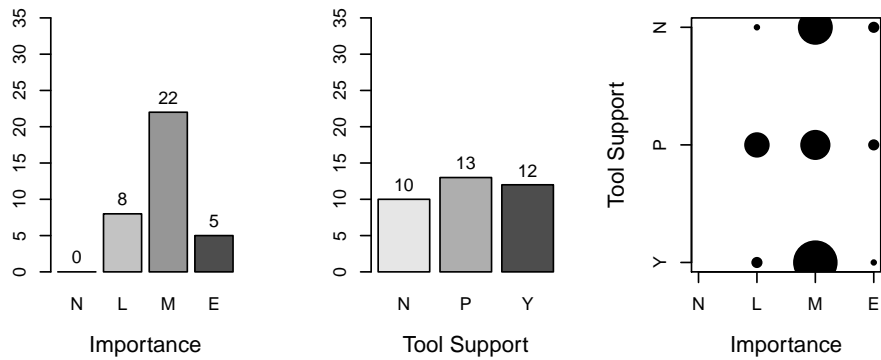


Figure 49: Did this class/method/function change (a lot) recently/in the past?

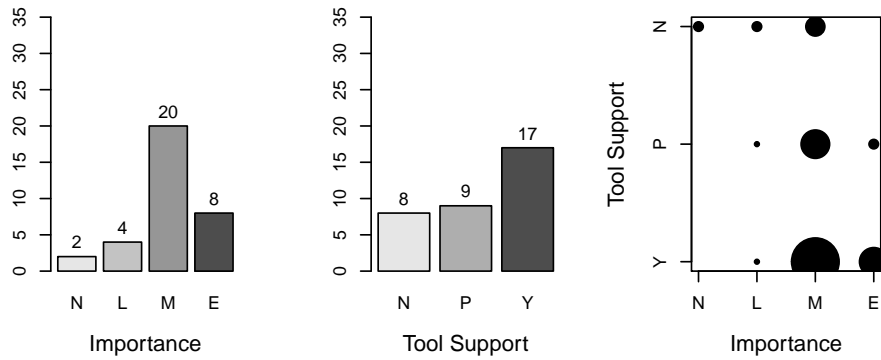


Figure 50: Is this change to a class/method/function the most recent one (in the branch)?

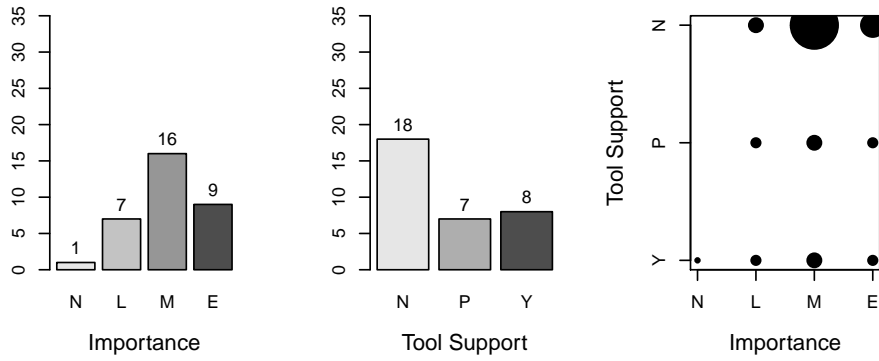


Figure 51: Is there any pending commit in the sequence of commits (in the branch) that might supersede this one?

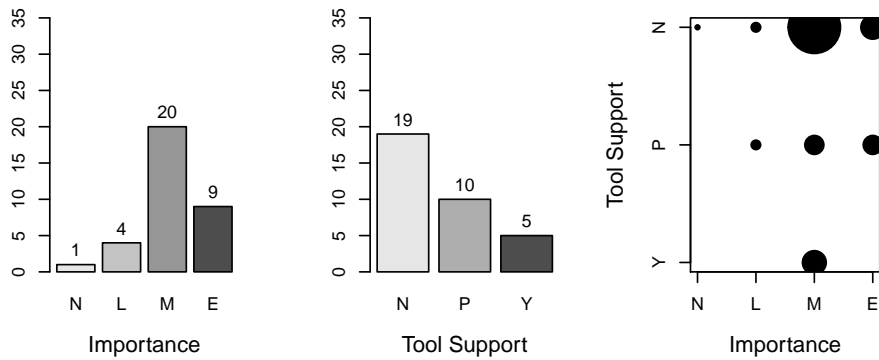


Figure 52: Is this commit part of a whole series of commits?

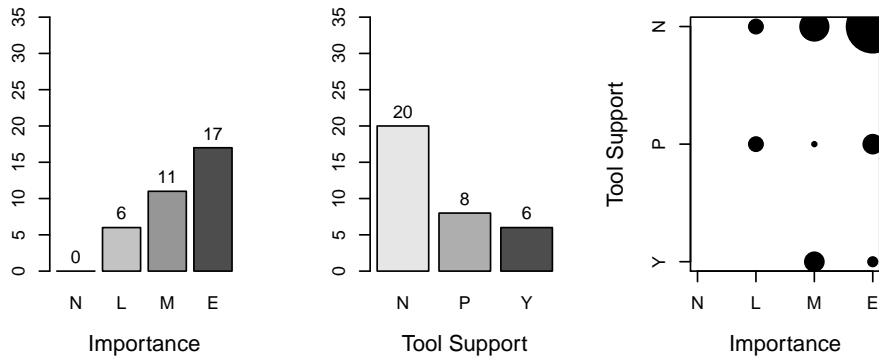


Figure 53: Does this commit depend on previous ones? (What are the other commits needed first to merge this commit?)

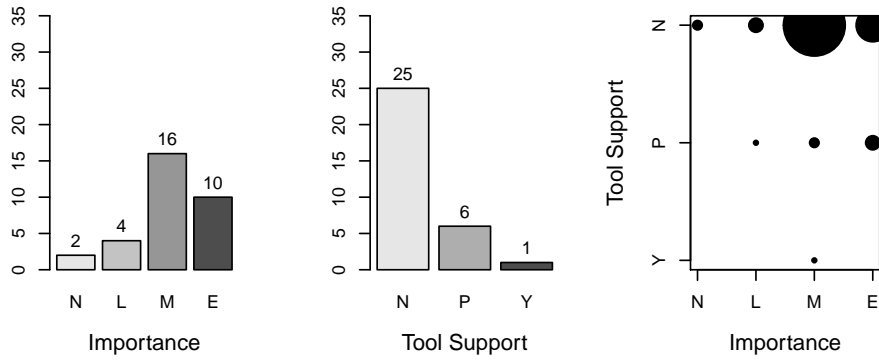


Figure 54: Is the change to a class/method/function ever used in subsequent changes?

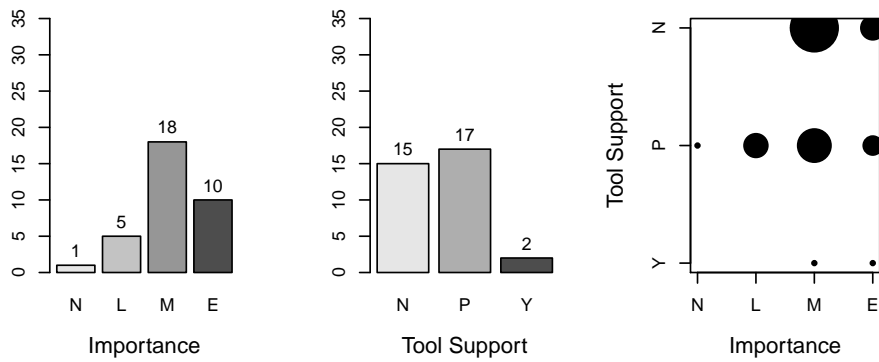


Figure 55: Is this change to a class/method/function reverting the code to an old state?

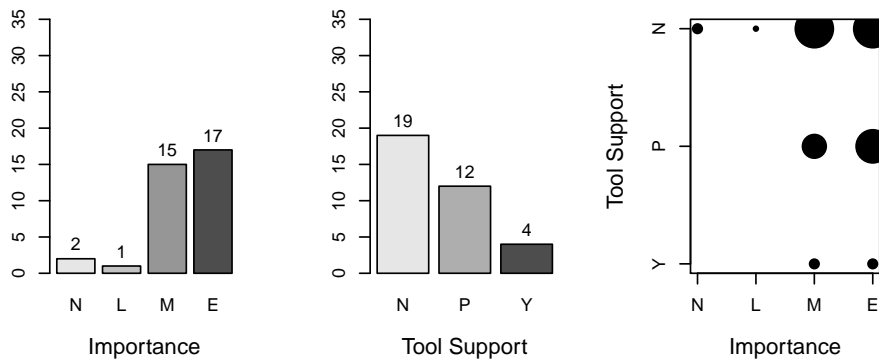


Figure 56: What else changed when this code was introduced or modified (i.e., documentation, website, database schema)?

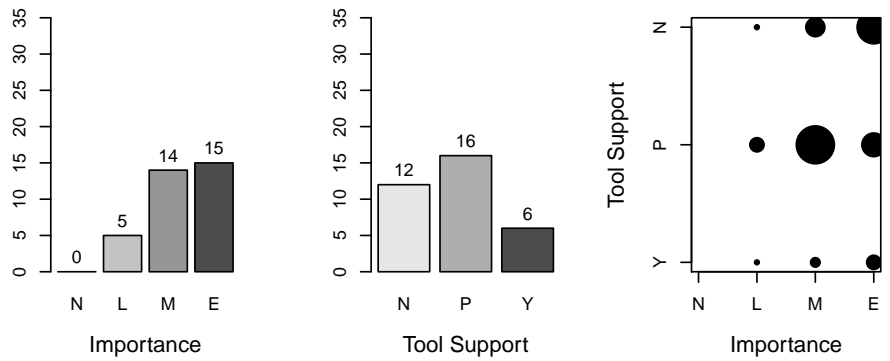


Figure 57: What other classes/methods/functions changed when this code was introduced or modified?

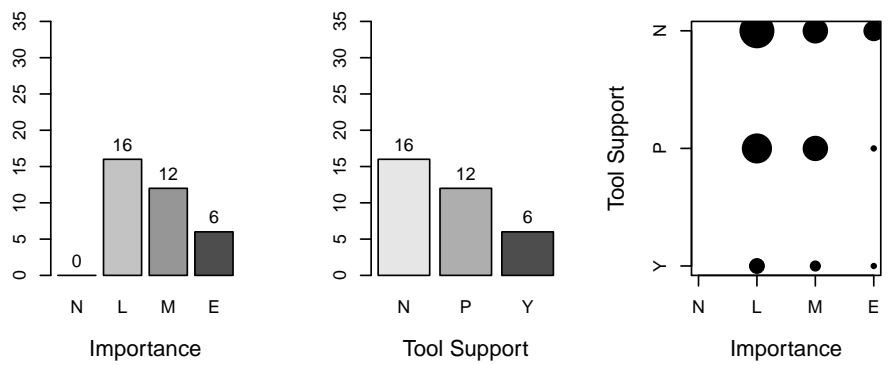


Figure 58: What are the other changes made by the same author/during the same time period?

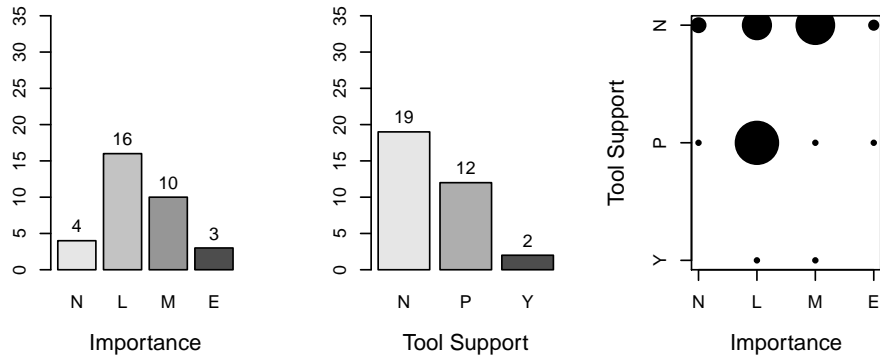


Figure 59: Did the changing classes/methods/functions of this commit change together in a previous commit?

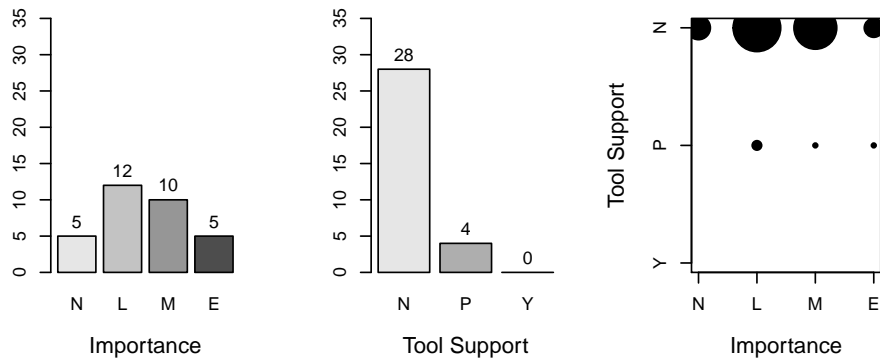


Figure 60: If there were changes to classes/methods/functions happening together in the past, can we suspect that there is still something missing in the current commit?

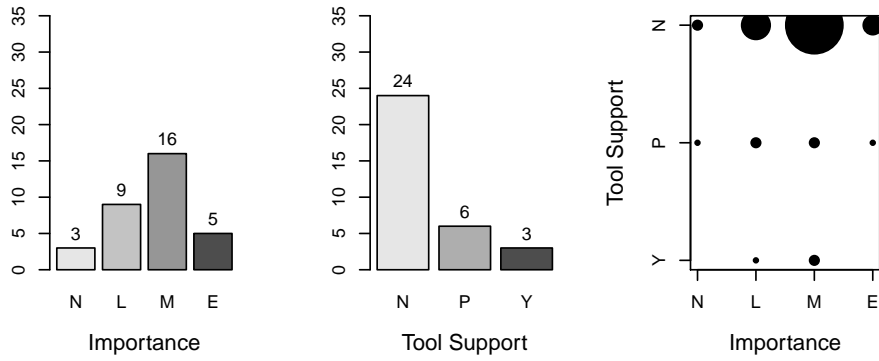


Figure 61: Were the classes/methods/functions affected by this change renamed in the past and if so, in which version of the system?

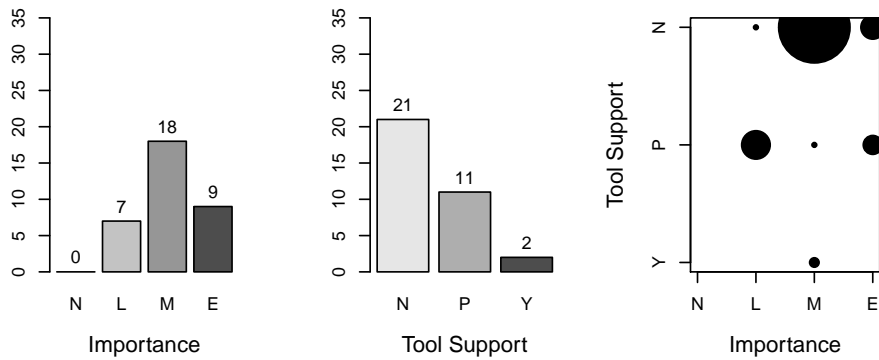


Figure 62: What were the users (callers) of a changed method/function in a particular version of the system?

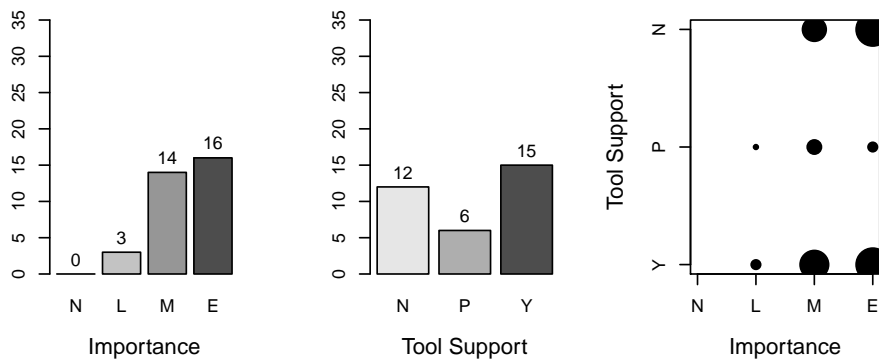


Figure 63: What are the current users (callers) of a changed method/function?

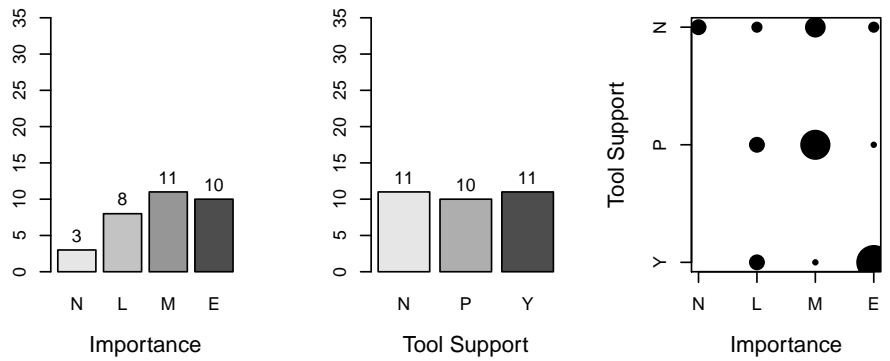


Figure 64: What commits of another branch have been integrated into this branch?

1. Are the changes in this commit consequence of some refactorings? In that case, can the refactoring be applied in the new conditions?
2. The time the developer spent on producing the fix / enhancement.
3. How much time/effort went into this change? Can I link a change to resources external to the change?
4. Too many questions, sorry.

Figure 65: Are there other question(s) that you ask when integrating changes and that we are missing in our catalogue?

5 Participant Profiles

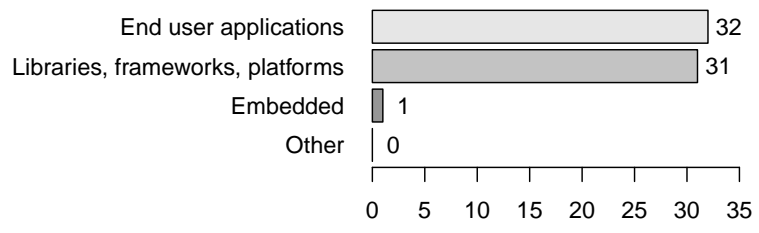


Figure 66: In what kind of software do you integrate changes from other developers?

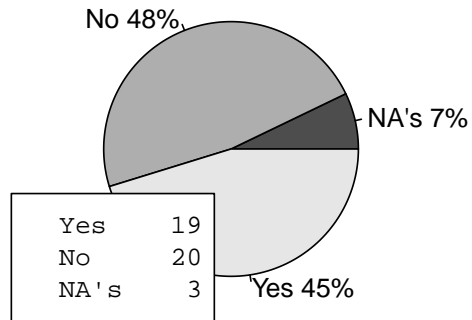


Figure 67: Is the project an open-source system?

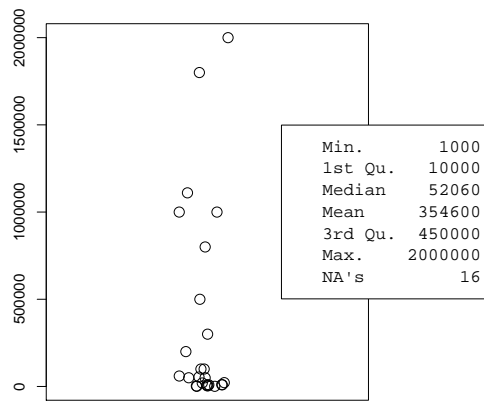


Figure 68: What is the approximated size of the system? (LOCs)

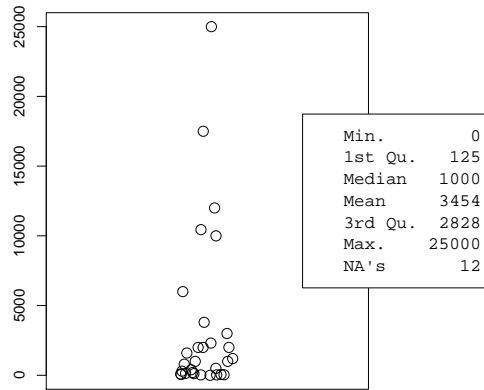


Figure 69: What is the approximated size of the system? (Classes)

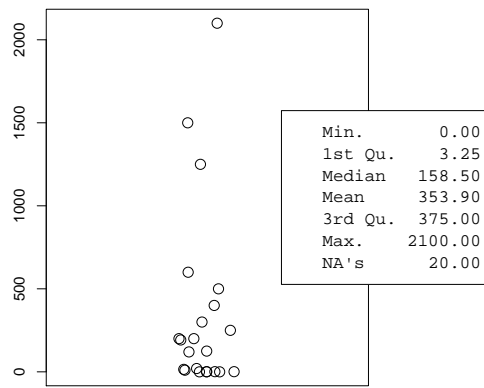


Figure 70: What is the approximated size of the system? (Files)

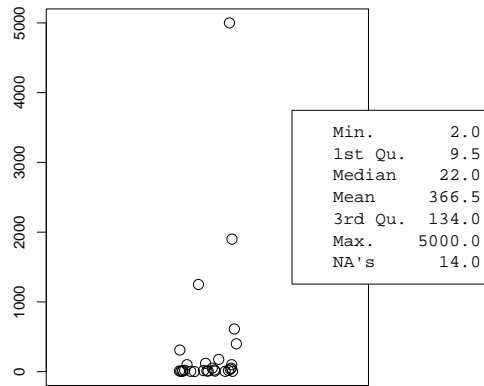


Figure 71: What is the approximated size of the system? (Packages)

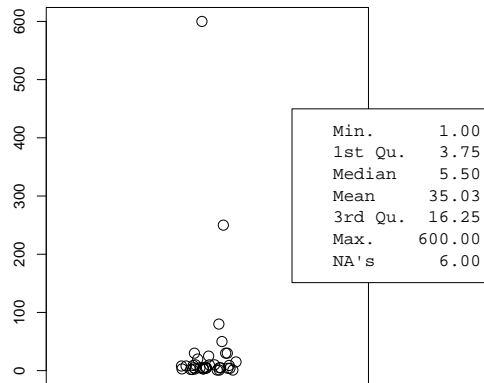


Figure 72: How many developers are working on this system?

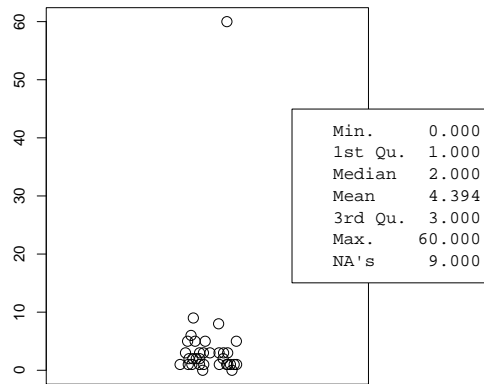


Figure 73: How many integrators are working on this system?

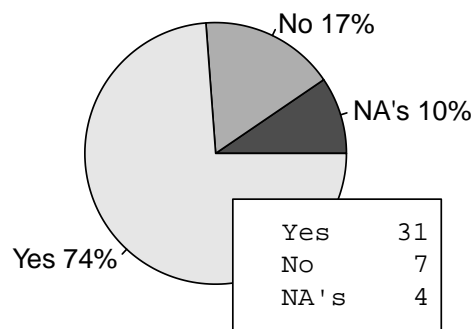


Figure 74: Are you currently one of the developers of this system? if YES, are you the main developer?

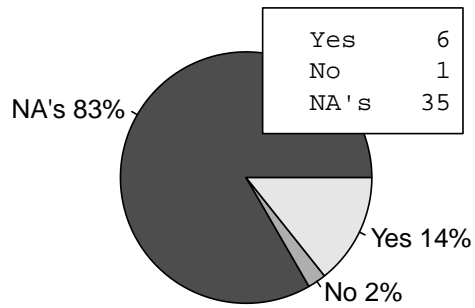


Figure 75: Are you currently one of the developers of this system? if NO, did you work previously on the system?

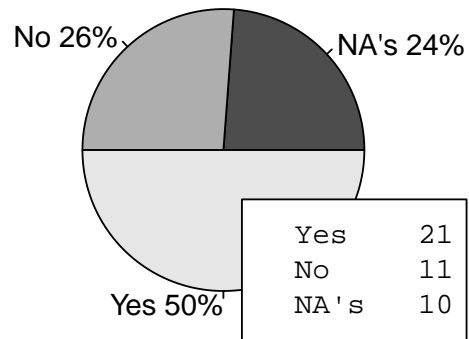


Figure 76: Are you the main integrator of this system? (Integrator by opposition to Developers)

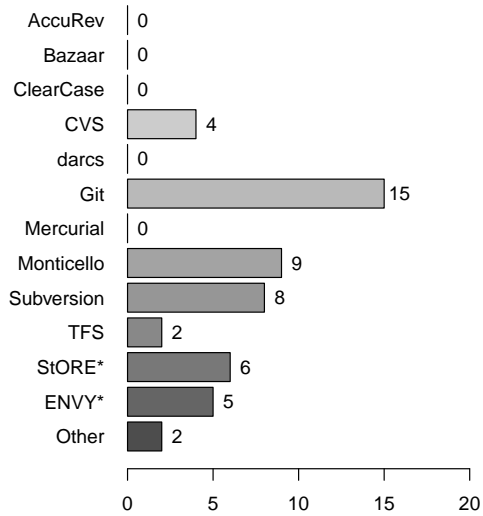


Figure 77: Which versioning tool(s) are using for this system? (categories with * were extracted from 'Other' field)

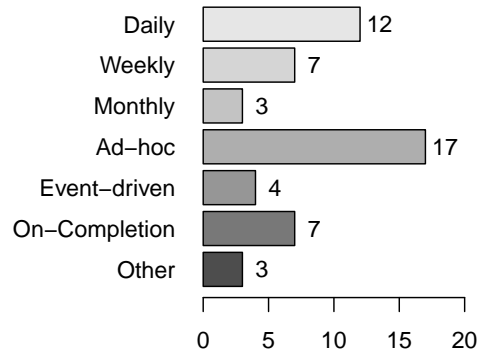


Figure 78: How often does the integration happen?

- | |
|--|
| <ol style="list-style-type: none"> 1. hourly 2. When something comes up 3. at every commit on the only SVN repo, plus every night |
|--|

Figure 79: How often does the integration happen? (Other)

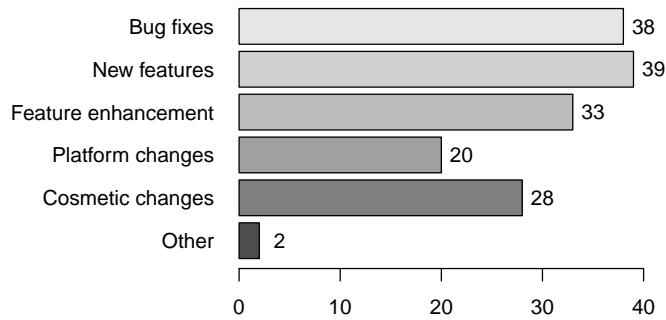


Figure 80: What is the nature of changes that are integrated?

- | |
|---|
| 1. database changes (change OO mapping) |
| 2. Refactoring |

Figure 81: What is the nature of changes that are integrated? (Other)

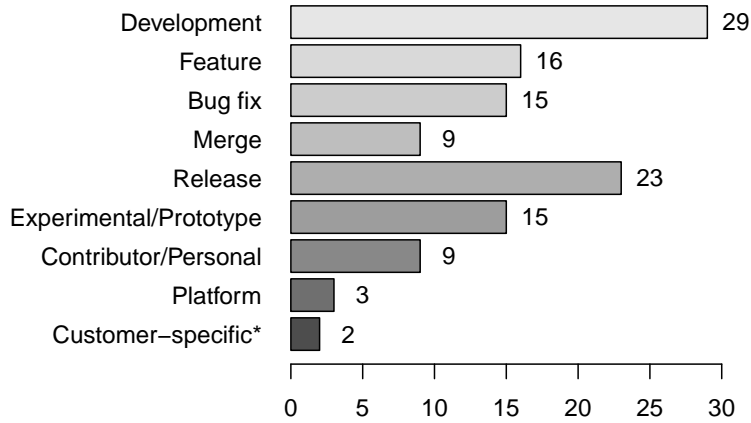


Figure 82: What types of branches are defined for this system? (categories with * were extracted from 'Other' field)

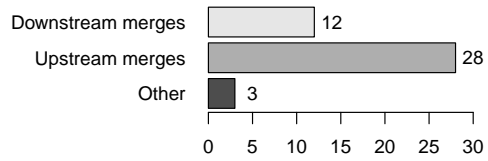


Figure 83: How do you merge changes?

1. From a feature branch to another feature branch.
2. I have a common platform, that is well sync.
3. There is not just a single direction for moving changes

Figure 84: How do you merge changes? (Other)

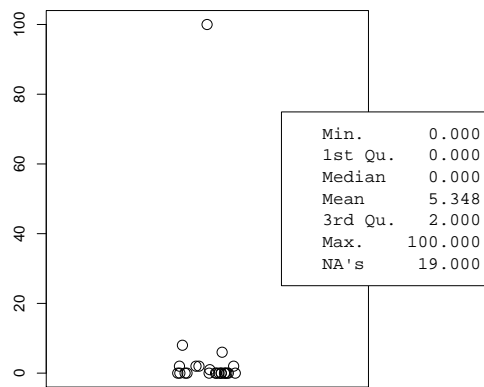


Figure 85: Does the system have forks? If so, how many?

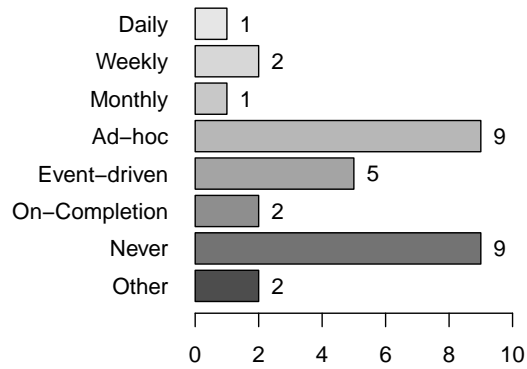


Figure 86: How often do you integrate/merge changes from a forked system?

- | |
|---|
| <ol style="list-style-type: none"> 1. There are no forks 2. When having pull requests |
|---|

Figure 87: How often do you integrate/merge changes from a forked system? (Other)

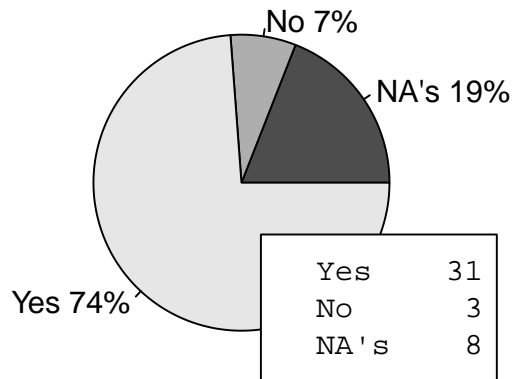


Figure 88: Do you interact with developers when integrating/merging changes? if yes, when in particular (eg. to solve conflicts, to provide you with information about changes, ...)?

1. Solve conflicts. Additional information from changes.
2. Solve conflicts. Ask motivation of strange changes. Code review feedback.
3. Solve conflicts. Provide info about changes.
4. When I am not happy about a change.
5. Asking for context. Resolving conflicts. Giving QA notes.
6. Understand changes.
7. Resolve conflicts. Discuss implementation issues.
8. If there're open questions, I simply communicate with the author to resolve them.
9. If the tool does not support the needed answers, I have to interact with other developers
10. To provide information about changes when it is incomplete
11. Really not very often as we know each other very well. So we need to interact only we're up to something quite unusual.
12. Indeed for acquiring information about the changes and solving conflicts if they occur
13. Mostly to clarify the changes you are merging with your own.
14. Solve conflicts
15. Mostly before the change to understand the reason behind the change. It also happens after the change when I disagree with the change.
16. Ask, create issues, re-iterate.

Figure 89: Do you interact with developers when integrating/merging changes? if yes, when in particular (eg. to solve conflicts, to provide you with information about changes, ...)? (Comments)

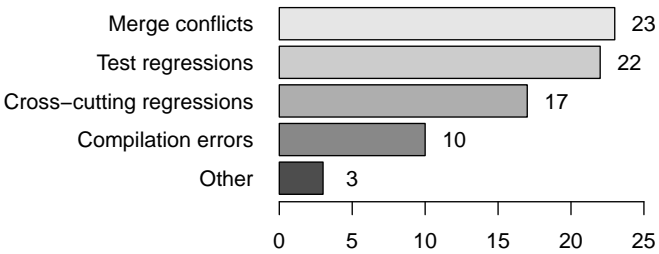


Figure 90: What are the most significant problems that you had have with merges?

1. When I am not happy about a change.
2. Performance regression
3. Telling people their code is not good enough

Figure 91: What are the most significant problems that you had have with merges? (Other)

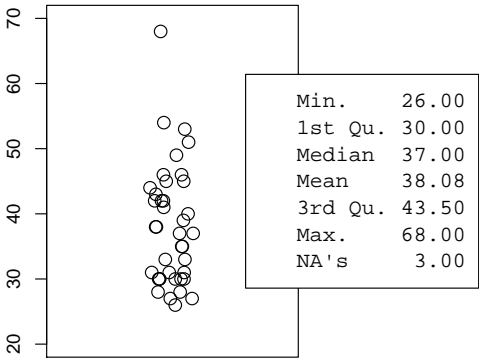


Figure 92: What is your age?

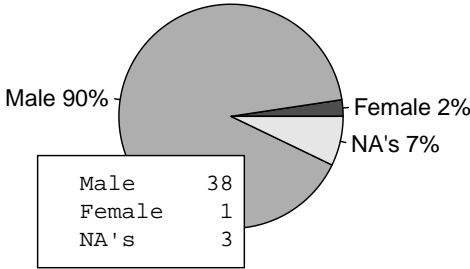


Figure 93: What is your gender?

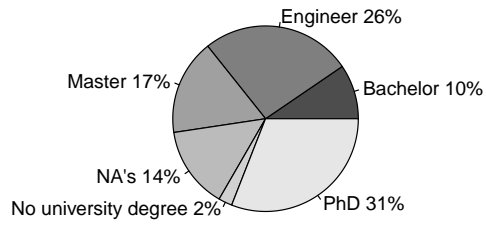


Figure 94: What is your academic background?

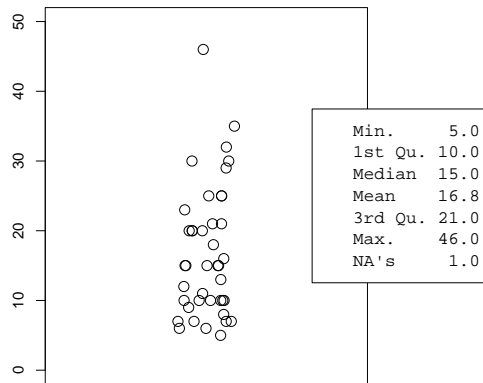


Figure 95: How many years have you been developing software?

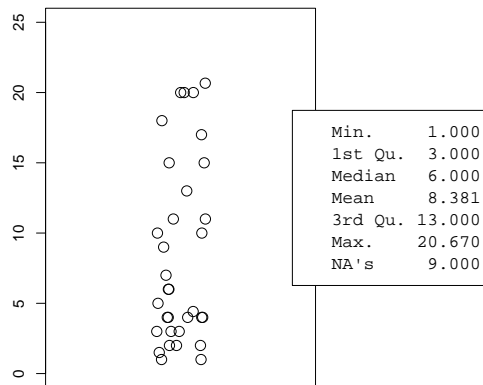


Figure 96: How long have you been integrating changes?

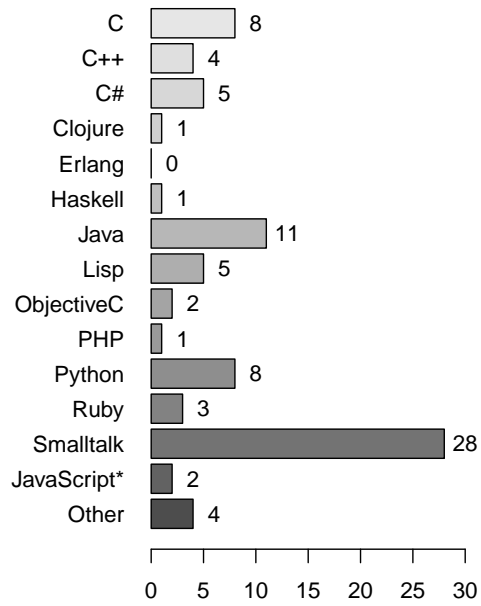


Figure 97: In which programming languages do you consider an expert? (categories with * were extracted from 'Other' field)