Code Reviews

Felipe Ebert (f.ebert@tue.nl)
Advisor: Fernando Castor (fjclf@cin.ufpe.br)
Co-Advisor: Alexander Serebrenik (a.serebrenik@tue.nl)

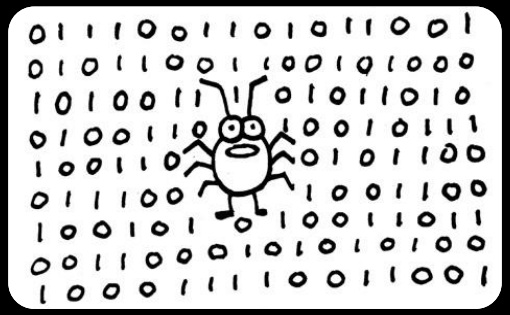
Understanding Confusion in Code Reviews

Felipe Ebert (f.ebert@tue.nl)
Advisor: Fernando Castor (fjclf@cin.ufpe.br)
Co-Advisor: Alexander Serebrenik (a.serebrenik@tue.nl)

Code Review

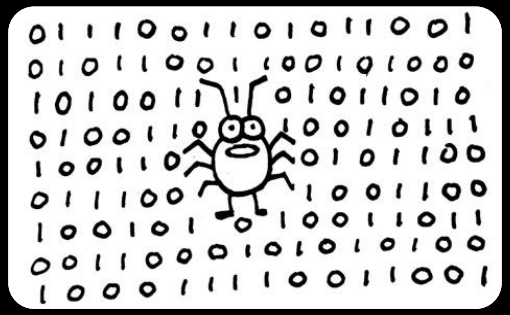
Who is doing Code Review?

Why Code Reviewing?!



Find Defects

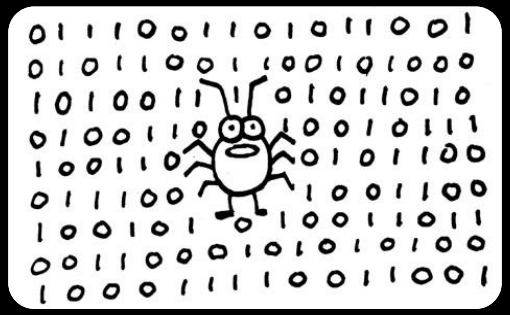
Why Code Reviewing?!



Find Defects

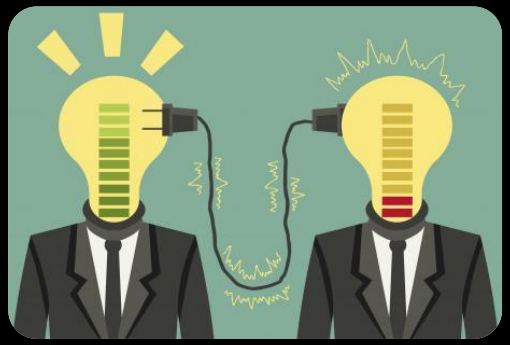
Code Improvement
Alternative Solutions

Why Code Reviewing?!



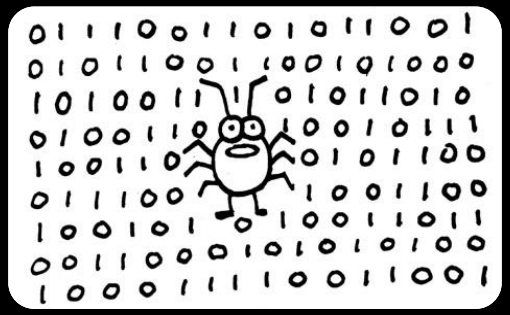
Find Defects

Code Improvement
Alternative Solutions



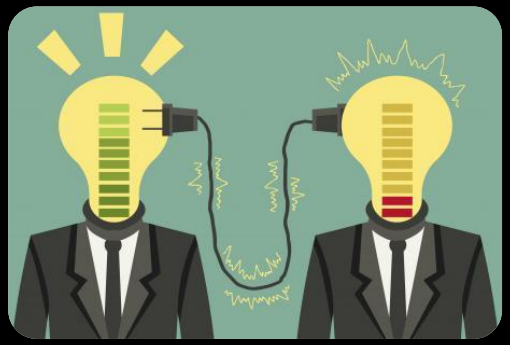
Knowledge Transfer

Why Code Reviewing?!



Find Defects

Code Improvement
Alternative Solutions

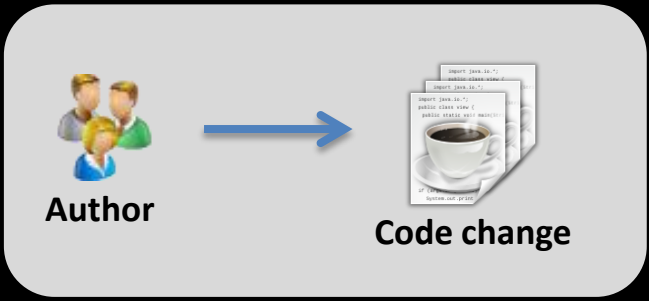


Knowledge Transfer

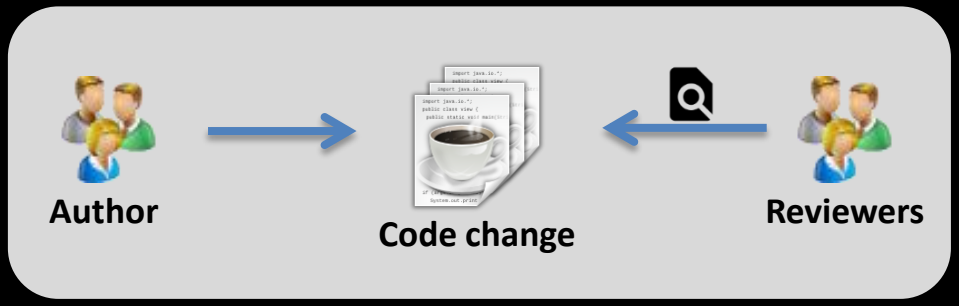
Team Awareness

Code Review Process

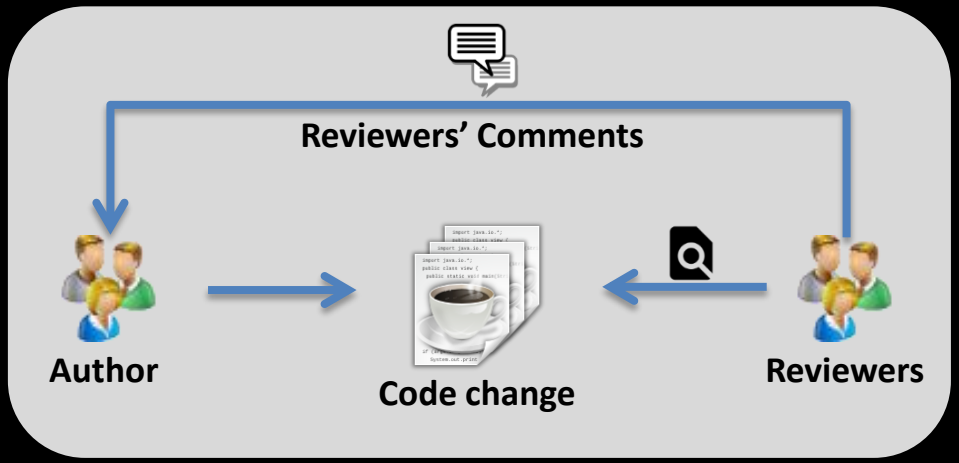
Code Review Process



Code Review Process



Code Review Process



General Comment

Dimitry Ivanov	Uploaded patch set 2.	Patchset 2	May 07, 2015 7:33 PM
Dimitry Ivanov	1 comment	Patchset 1	May 07, 2015 7:36 PM
Elliott Hughes	Code-Review +2	Patchset 2	May 07, 2015 7:37 PM
Serban Constantinescu	Added to reviewer: Serban Constantinescu		May 07, 2015 7:45 PM

Serban Constantinescu — Patchset 2 — May 07, 2015 7:45 PM

I am not too familiar with this code. But the code that walks the init_array and fini_array seems architecture agnostic. Thus I would be in favour of removing the ifndef __ARM__.

I suspect that magic was added there for marking the end of the init/fini_array. But now we seem to calculate the fini and init array sizes using:

```
init_array_count_ = static_cast<uint32_t>(d->d_un.d_val) / sizeof(ElfW(Addr));

fini_array_count_ = static_cast<uint32_t>(d->d_un.d_val) / sizeof(ElfW(Addr));
```

Snippets from linker/linker.cpp

Inline Comment

49682 Properly align init/fini_arrays for crtend.o libc/arch-common/bionic/crtend.S File 3 of 4 Prev Up Next

Base gitiles → Patchset 1 gitiles DOWNLOAD SHOW BLAME Diff view:

```
30
31 »       .section .preinit_array, "aw"
32 »       ASM_PTR_SIZE(0)
33
34 »       .section .init_array, "aw"
35 »       ASM_PTR_SIZE(0)
36
37 »       .section .fini_array, "aw"
38 »       ASM_PTR_SIZE(0)
39
40 #if defined(__linux__) && defined(__ELF__)
41 »       .section .note.GNU-stack,"",%progbits
42 #endif
43 #if defined(__i386__) || defined(__x86_64__)
44 »       .section»       .eh_frame,"a",@progbits
45 »       ASM_ALIGN(4)
```

```
30
31 »       .section .preinit_array, "aw"
32 »       ASM_ALIGN_TO_PTRSIZE
33 »       ASM_PTR_SIZE(0)
34
35 »       .section .init_array, "aw"
36 »       ASM_ALIGN_TO_PTRSIZE
37 »       ASM_PTR_SIZE(0)
38
39 »       .section .fini_array, "aw"
40 »       ASM_ALIGN_TO_PTRSIZE
41 »       ASM_PTR_SIZE(0)
42
43 #if defined(__linux__) && defined(__ELF__)
44 »       .section .note.GNU-stack,"",%progbits
45 #endif
46 #if defined(__i386__) || defined(__x86_64__)
47 »       .section»       .eh_frame,"a",@progbits
48 #if defined(__i386__)
49 »       .balign 4
```

Dimitry Ivanov May 07, 2015

This was preserved from previous state.. the question is do we really need to align eh_frame segment. And if we do, why only for i386.

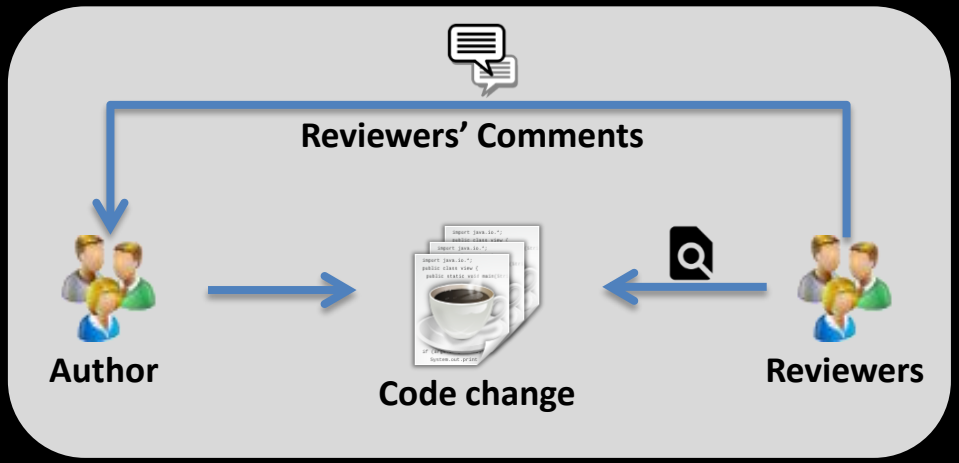
Serban Constantinescu May 07, 2015

Done

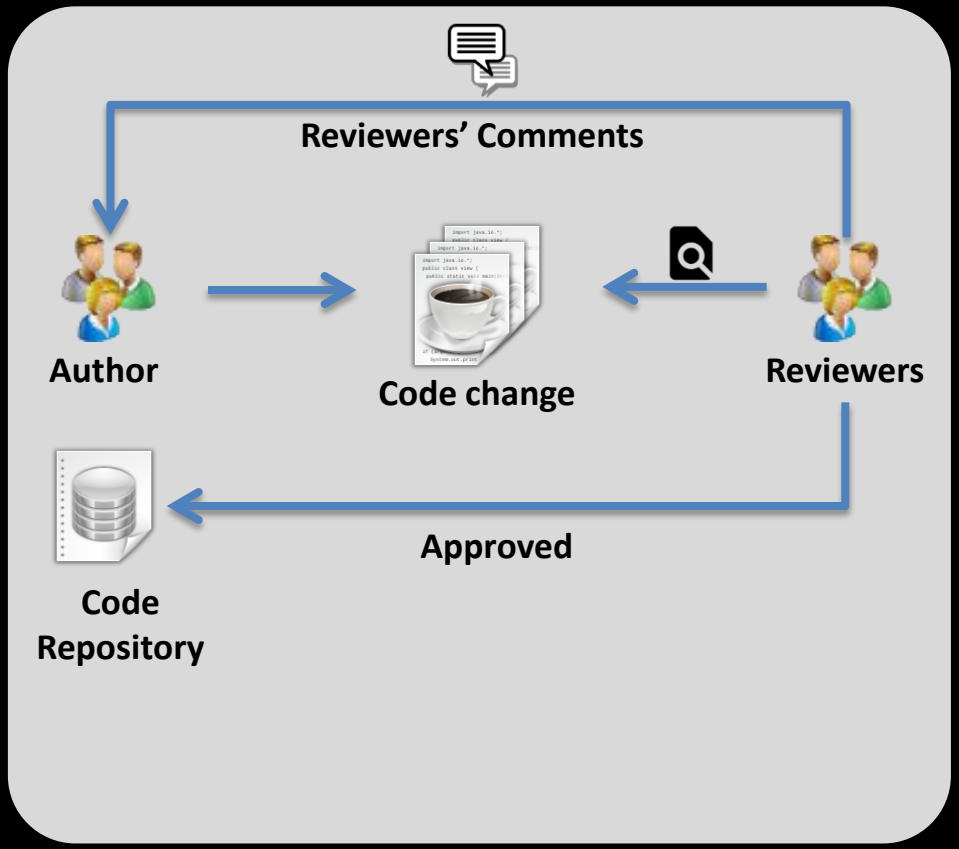
REPLY QUOTE ACK DONE

```
50 #endif
```

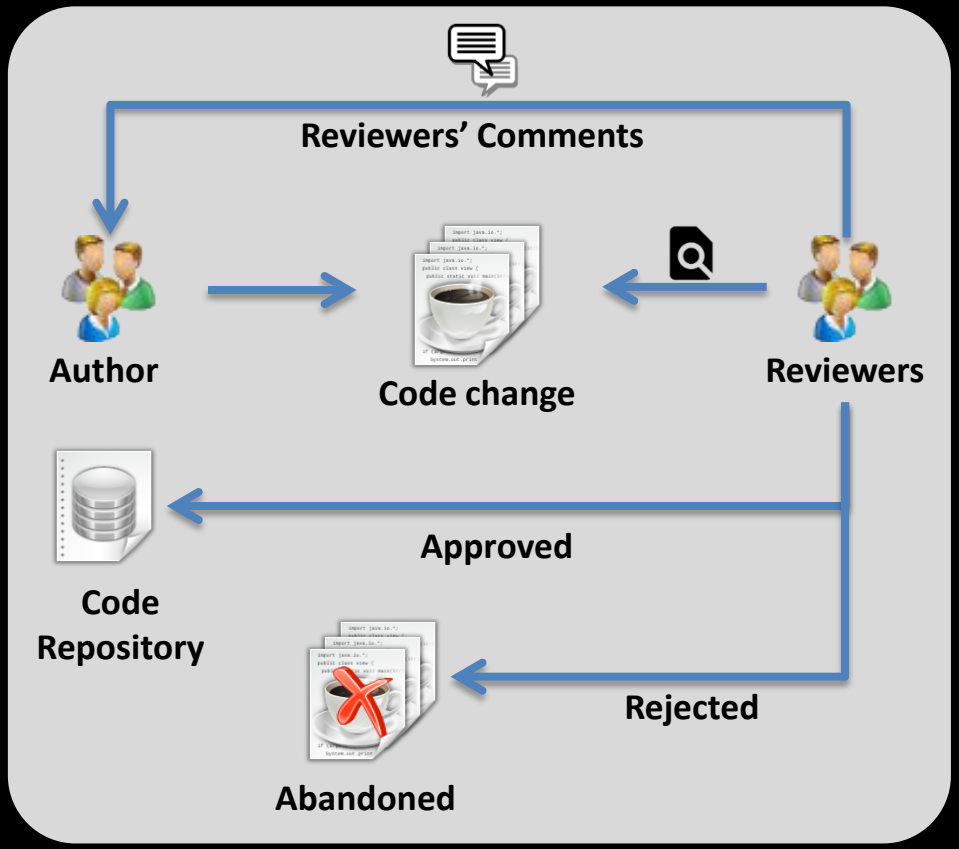
Code Review Process



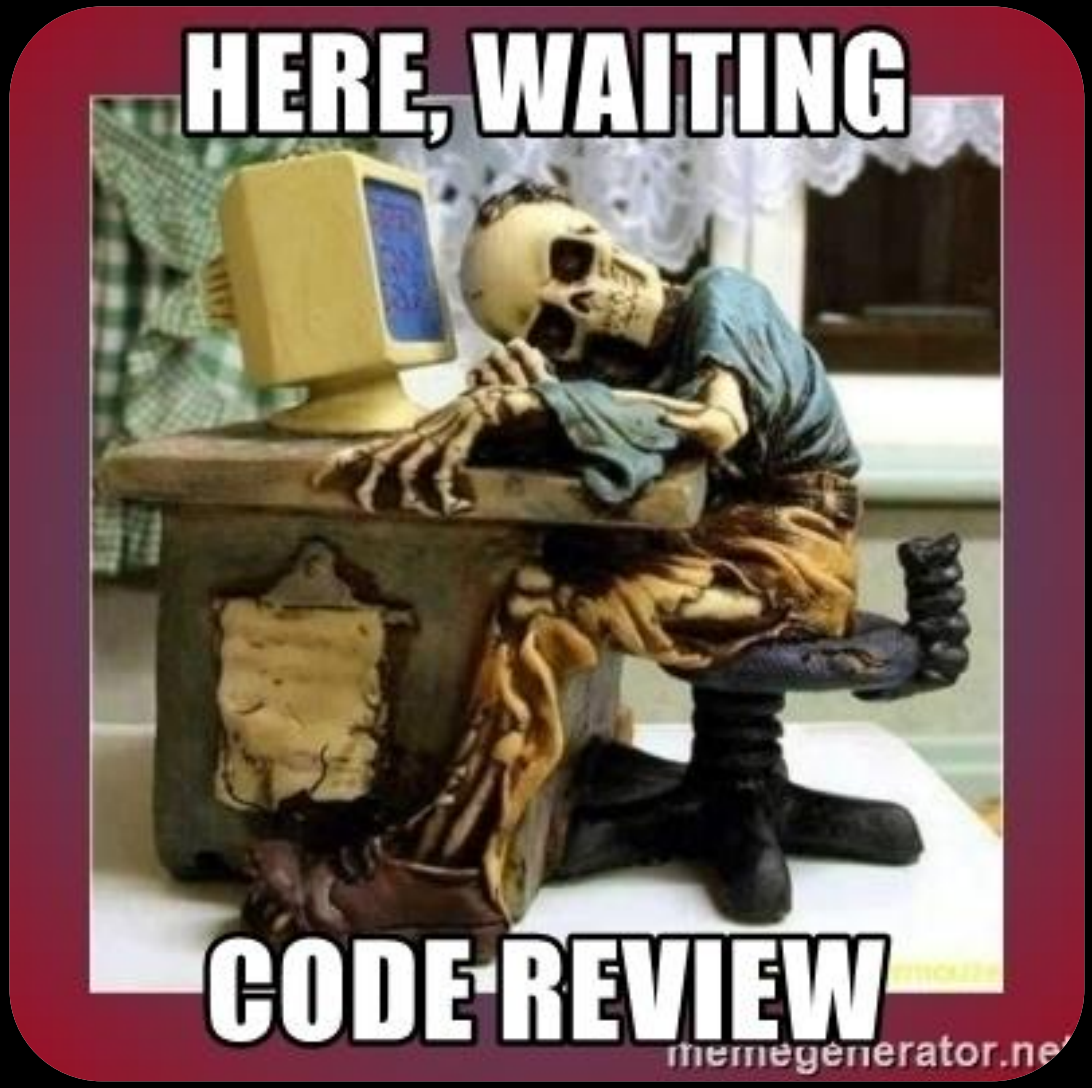
Code Review Process



Code Review Process



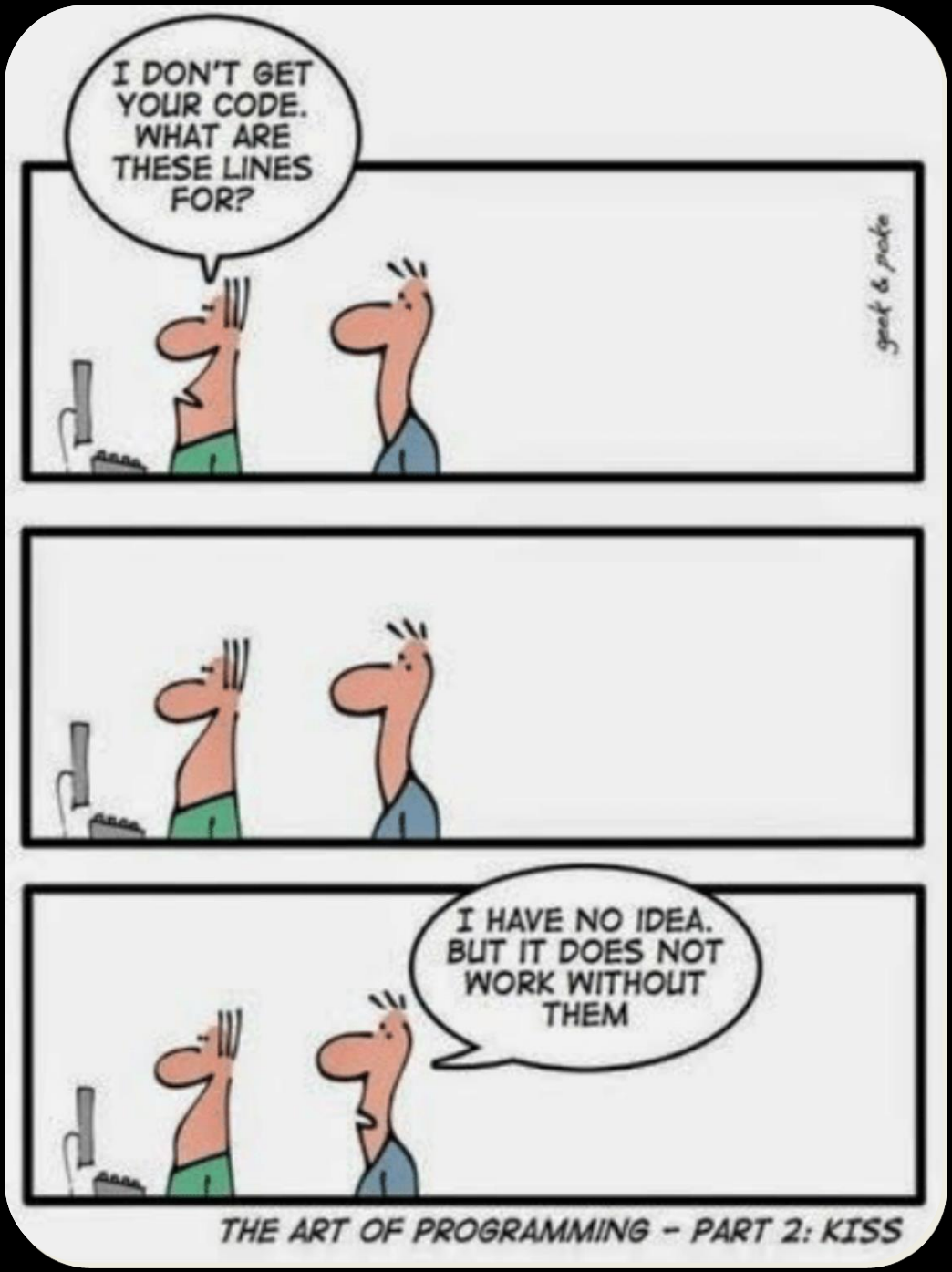
Code Reviews Are Not Free!!!



To study confusion in code reviews, its manifestations, causes, and impacts

- Lack of knowledge
- Lack of tools

Why confusion?!

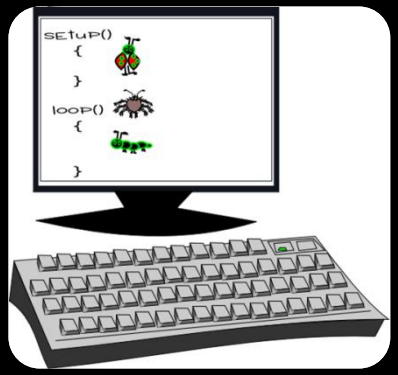


What is confusion?!

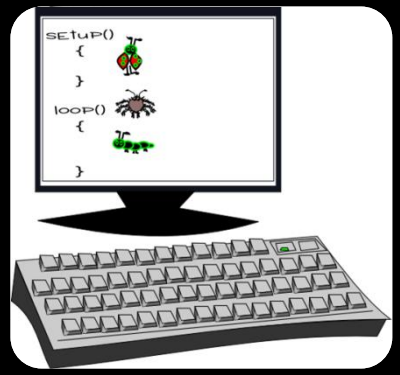
a situation where a person is **uncertain** about or **unable to understand** something

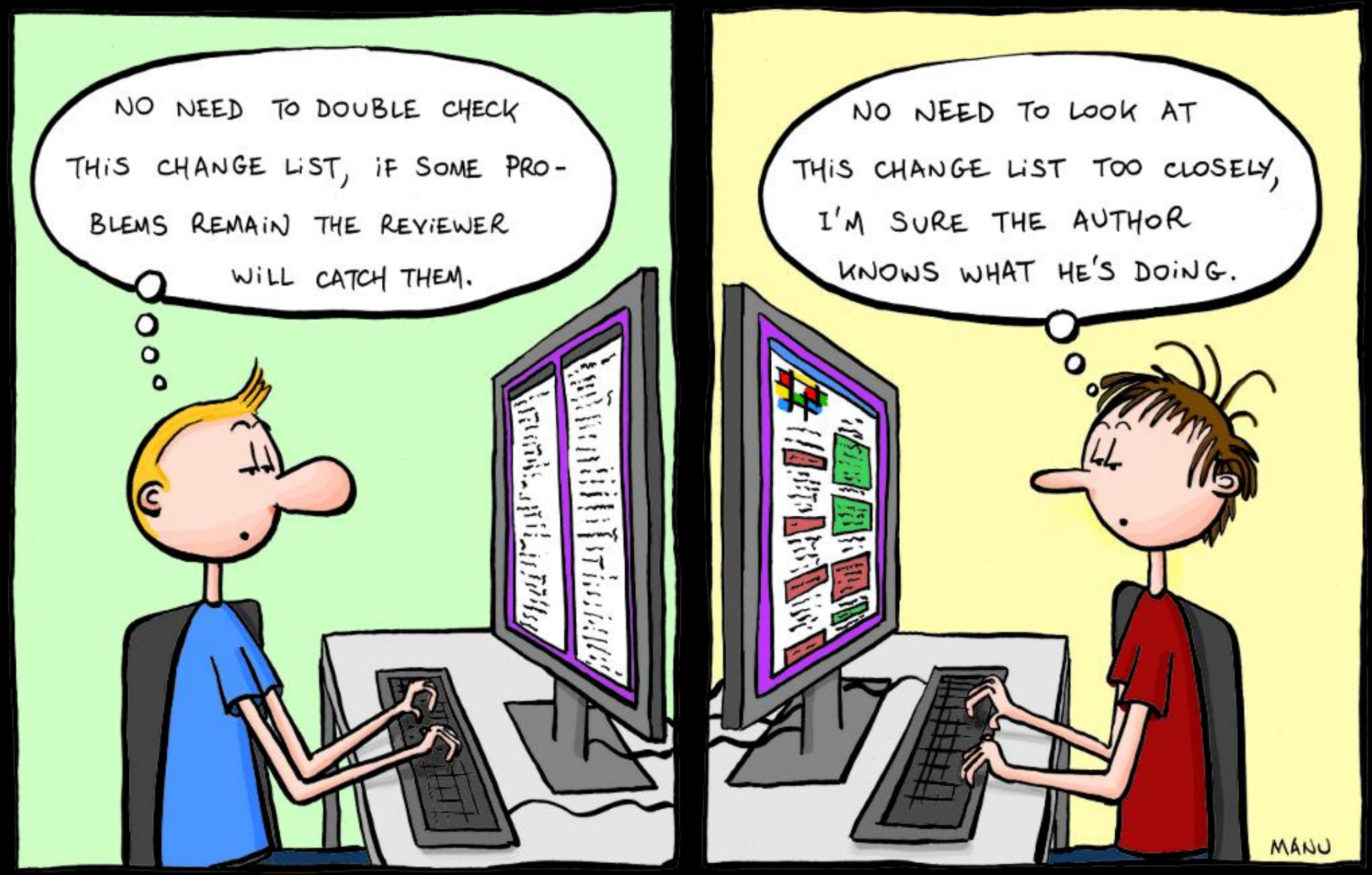
The impacts of confusion!

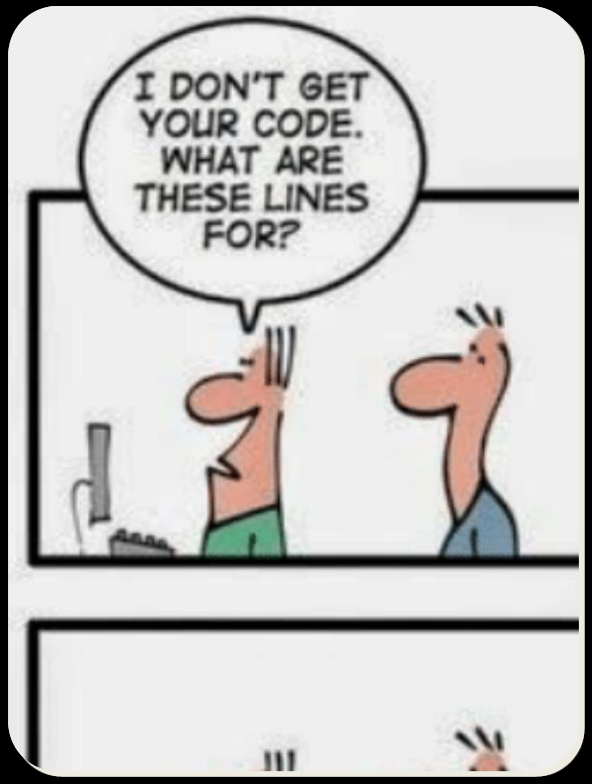
The impacts of confusion!



The impacts of confusion!




NO NEED TO DOUBLE CHECK THIS CHANGE LIST, IF SOME PROBLEMS REMAIN THE REVIEWER WILL CATCH THEM.
NO NEED TO LOOK AT THIS CHANGE LIST TOO CLOSELY, I'M SURE THE AUTHOR KNOWS WHAT HE'S DOING.
MANU



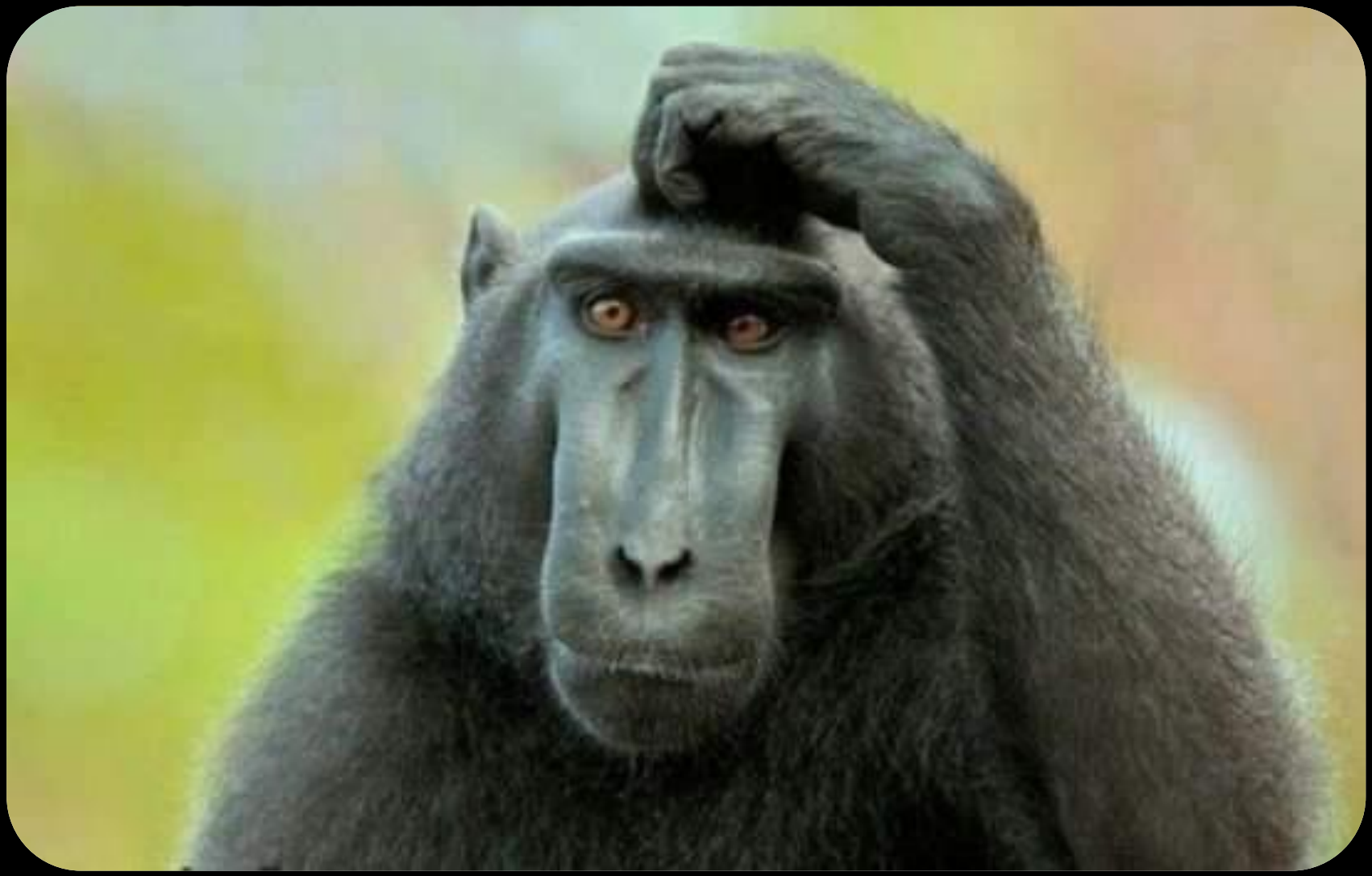
I'm unsure *as to whether the loop is necessary here for the same reason as above, or whether the array actually needs to be reset to handle disk changes.*

Patch Set 2: Code-Review+2

Though ***I don't really understand*** *why ValueObject moved to runtime...*

Patch Set 1:

What's the context? *Is this fixing/improving existing code? Could you use the assembler tests for it?*



How to identify confusion?

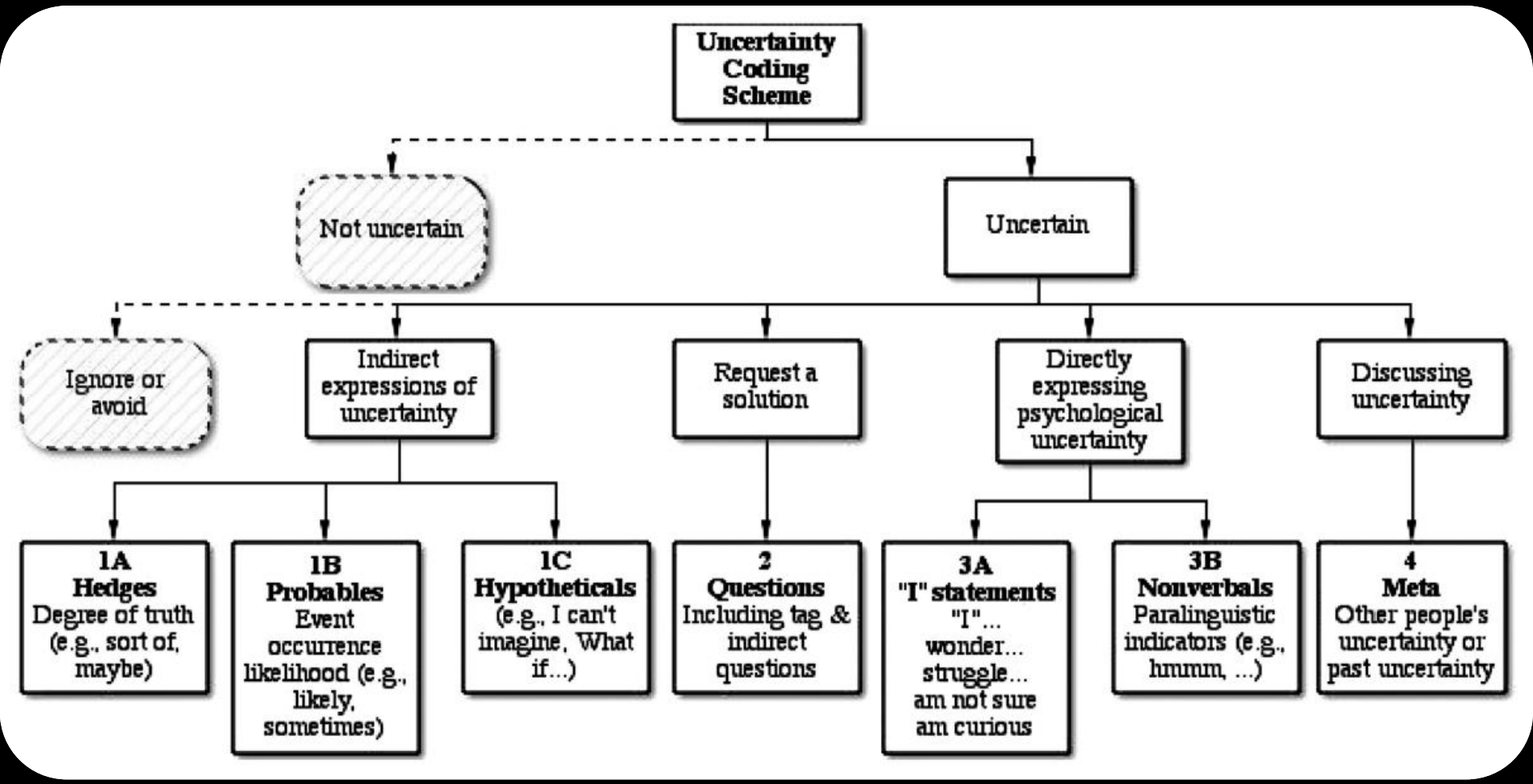
Confusion Detection in Code Reviews

1st Study

EBERT, F.; CASTOR, F.; NOVIELLI, N.; SEREBRENIK, A. Confusion detection in code reviews. In: 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME). Shanghai, China: IEEE Computer Society, 2017. p. 549–553.

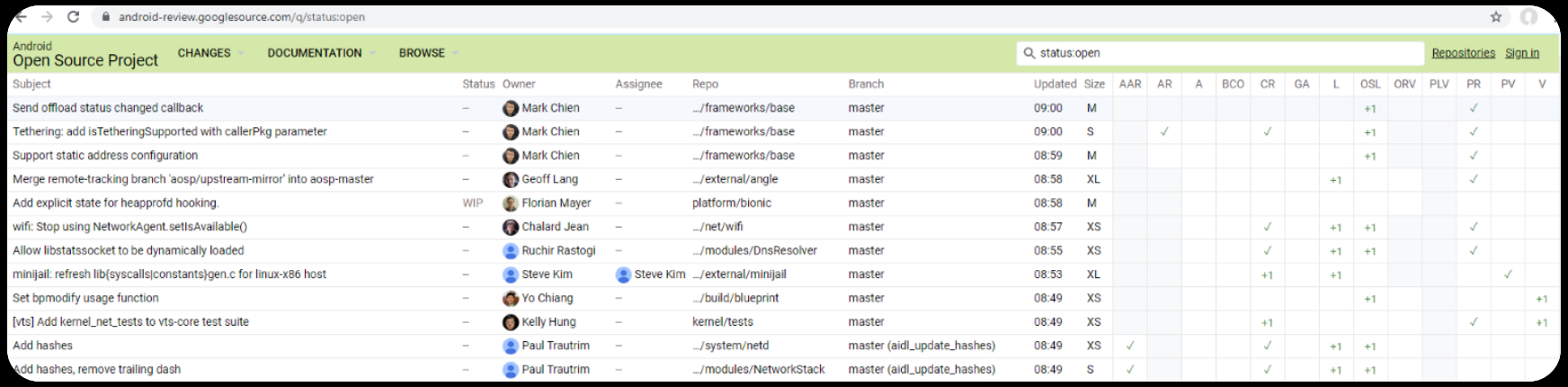
Research Questions

- RQ1: Can human raters agree on the presence of confusion in code review comments?
- RQ2: Is it possible to design a tool to recognize confusion in developers' comments?



Michelle E. Jordan et al., "Expressing uncertainty in computer-mediated discourse: Language as a marker of intellectual work," Discourse Processes, vol. 49, no. 8, pp. 660–692, 2012.

Gerrit Code Review



Subject	Status	Owner	Assignee	Repo	Branch	Updated	Size	AAR	AR	A	BCO	CR	GA	L	OSL	ORV	PLV	PR	PV	V
Send offload status changed callback	–	Mark Chien	–	.../frameworks/base	master	09:00	M								+1			✓		
Tethering: add isTetheringSupported with callerPkg parameter	–	Mark Chien	–	.../frameworks/base	master	09:00	S		✓			✓			+1			✓		
Support static address configuration	–	Mark Chien	–	.../frameworks/base	master	08:59	M								+1			✓		
Merge remote-tracking branch 'aosp/upstream-mirror' into aosp-master	–	Geoff Lang	–	.../external/angle	master	08:58	XL							+1				✓		
Add explicit state for heapprofd hooking.	WIP	Florian Mayer	–	platform/bionic	master	08:58	M													
wifi: Stop using NetworkAgent.setIsAvailable()	–	Chalard Jean	–	.../net/wifi	master	08:57	XS					✓		+1	+1			✓		
Allow libstatssocket to be dynamically loaded	–	Ruchir Rastogi	–	.../modules/DnsResolver	master	08:55	XS					✓		+1	+1			✓		
minijail: refresh lib{syscalls|constants}gen.c for linux-x86 host	–	Steve Kim	Steve Kim	.../external/minijail	master	08:53	XL					+1		+1					✓	
Set bpmodify usage function	–	Yo Chiang	–	.../build/blueprint	master	08:49	XS								+1					+1
[vts] Add kernel_net_tests to vts-core test suite	–	Kelly Hung	–	kernel/tests	master	08:49	XS					+1						✓		+1
Add hashes	–	Paul Trautrim	–	.../system/netd	master (aidl_update_hashes)	08:49	XS	✓				✓		+1	+1					
Add hashes, remove trailing dash	–	Paul Trautrim	–	.../modules/NetworkStack	master (aidl_update_hashes)	08:49	S	✓				✓		+1	+1					

gerrit-review.googlesource.com/Documentation/rest-api.html

Table of Contents

Gerrit Code Review - REST API

Version V3.1.3-1828-G9eb72e8526 |

Gerrit Code Review comes with a REST like API available over HTTP. The API is suitable for automated tools to build upon, as well as supporting some ad-hoc scripting use cases.

See also: REST API Developers' Notes.

Endpoints

/access/

Access Right related REST endpoints

/accounts/

Account related REST endpoints

/changes/

Change related REST endpoints

/config/

Config related REST endpoints

/groups/

Group related REST endpoints

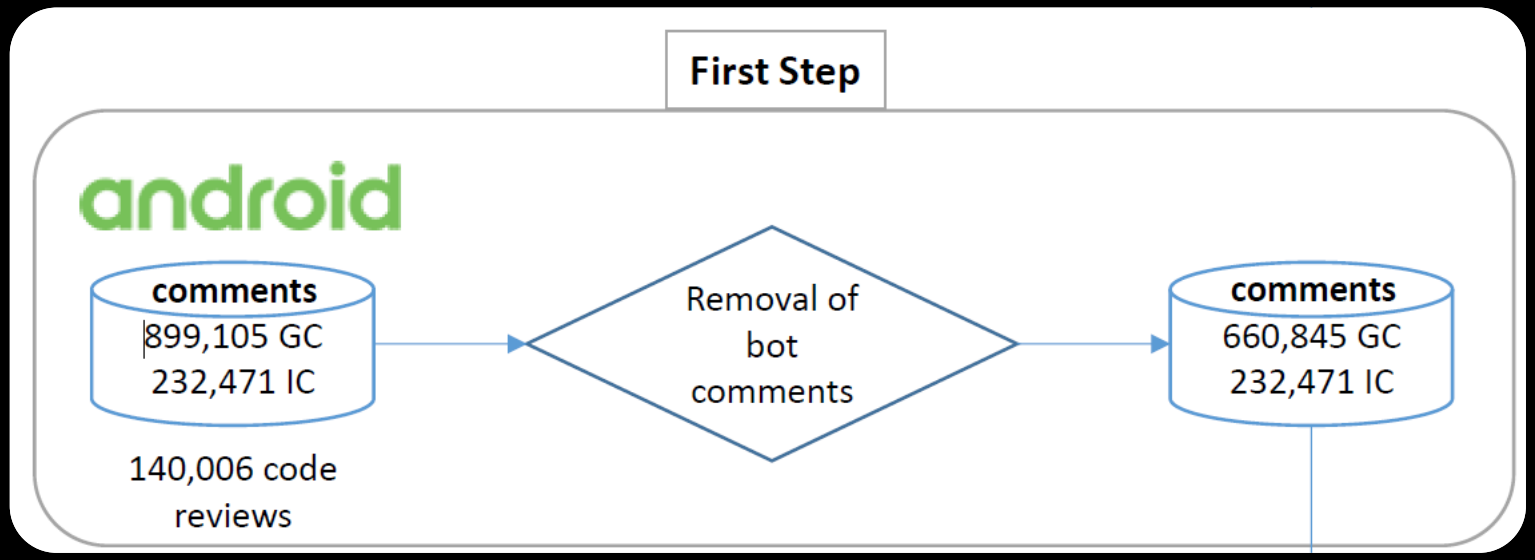
/plugins/

Plugin related REST endpoints

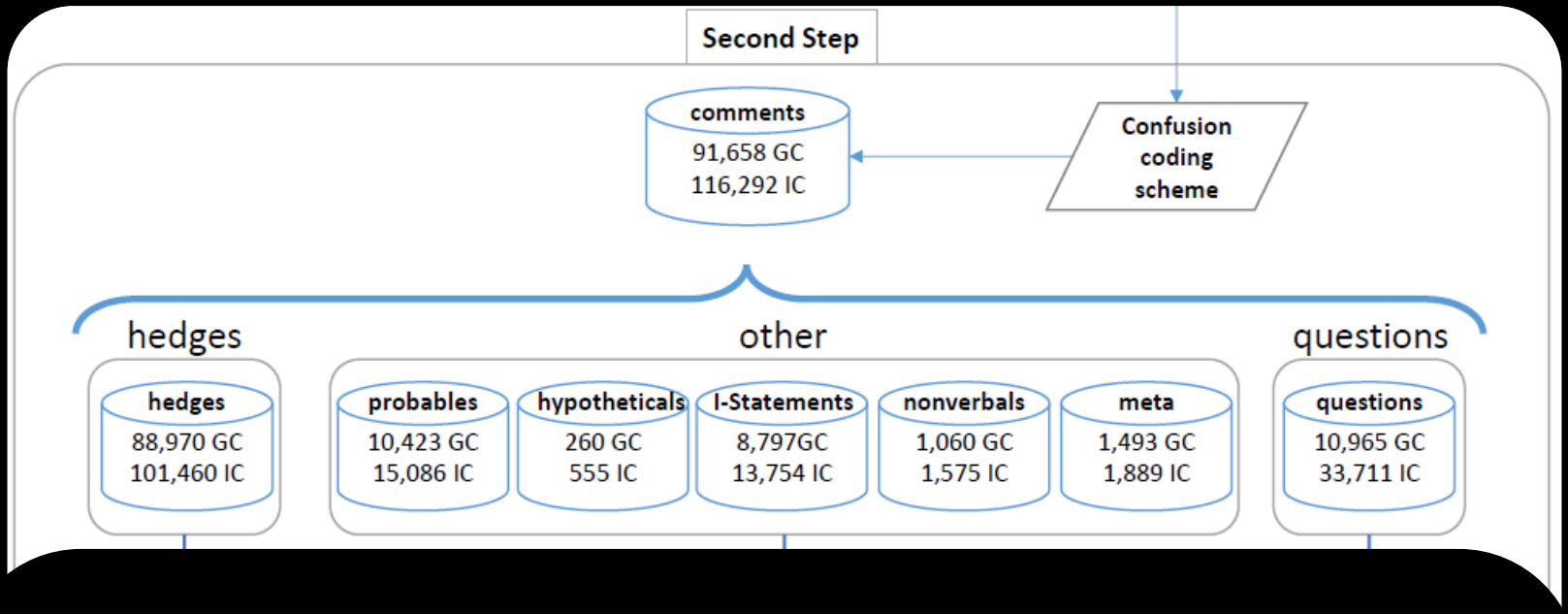
/projects/

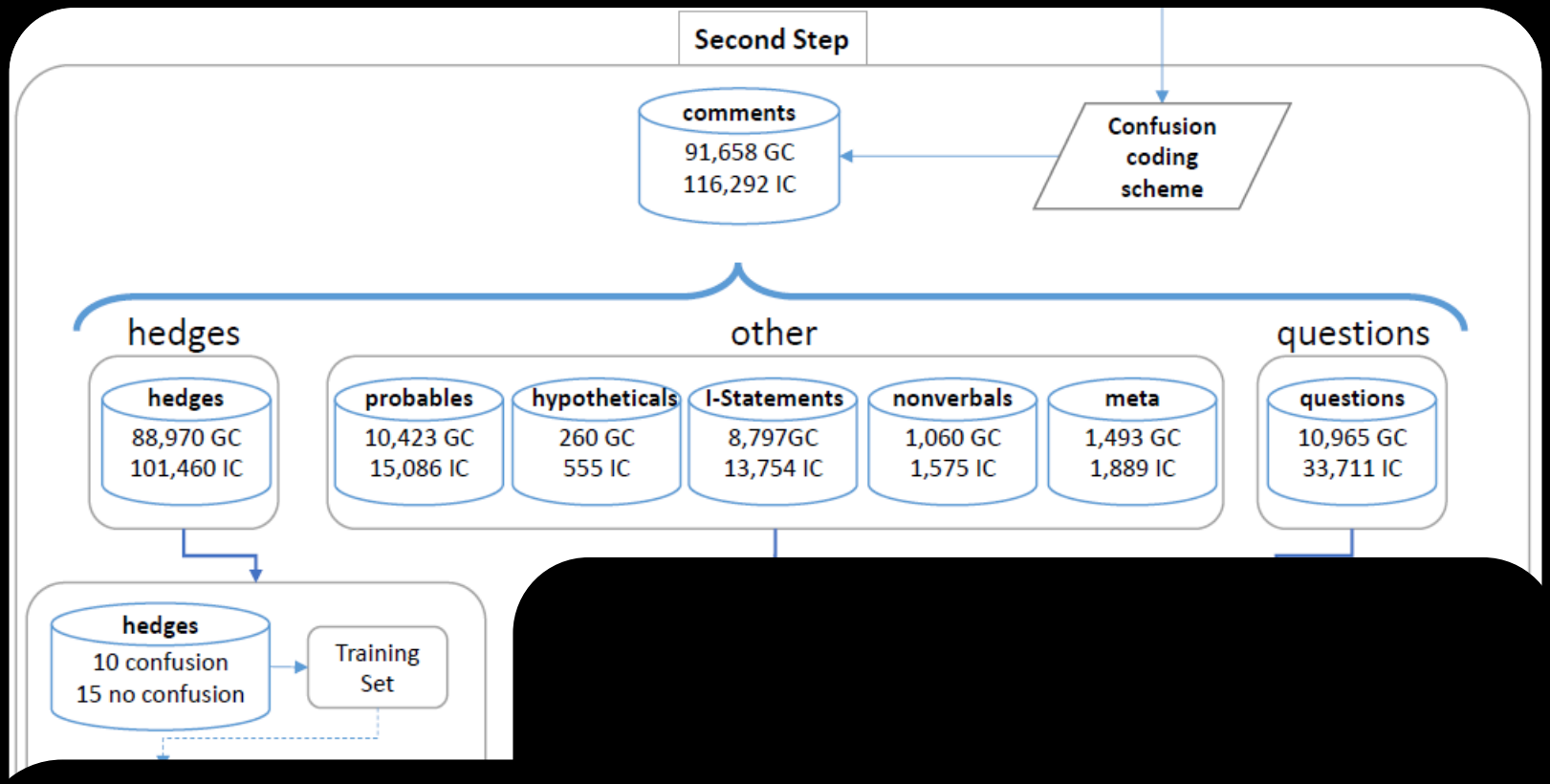
Project related REST endpoints

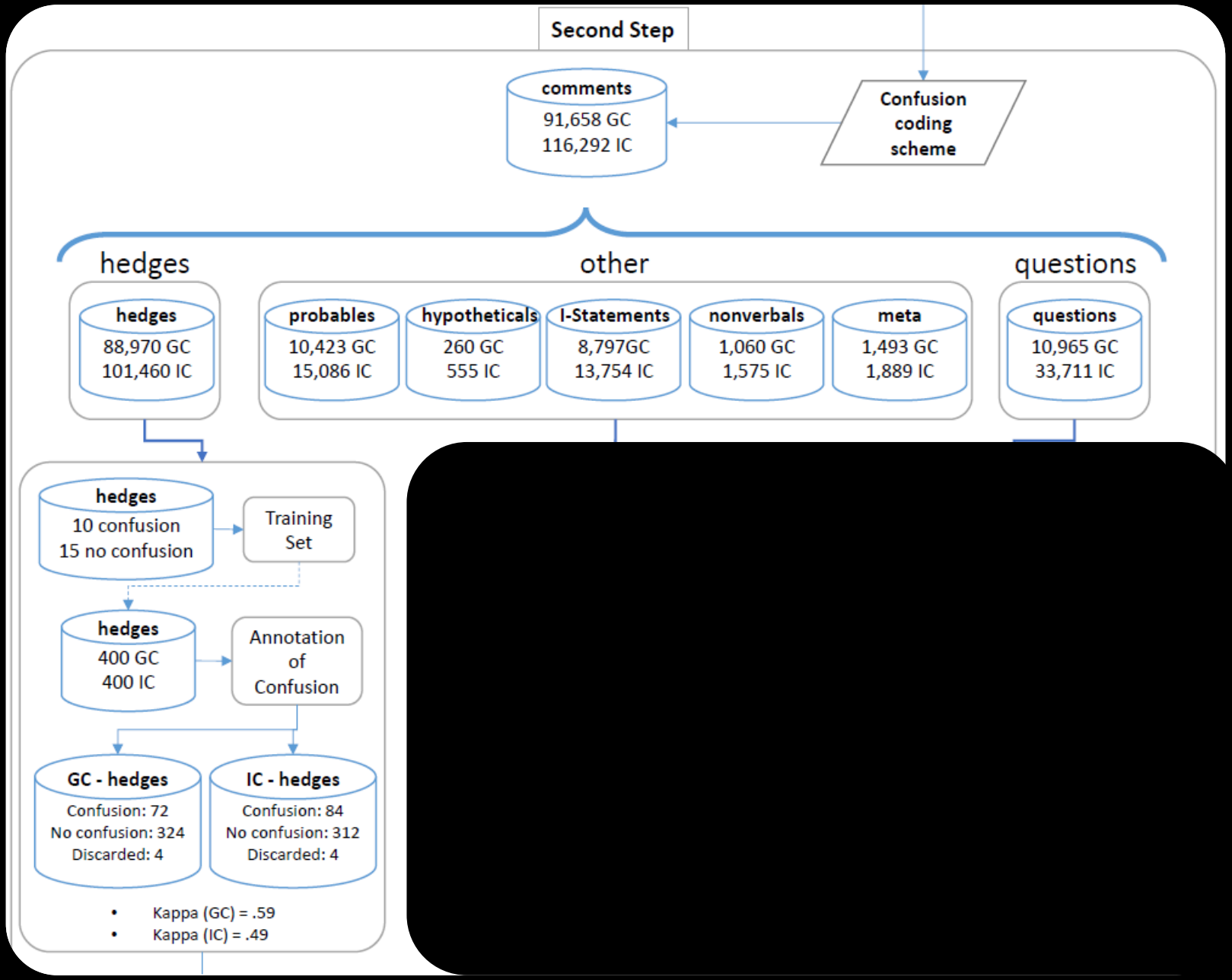
/Documentation/

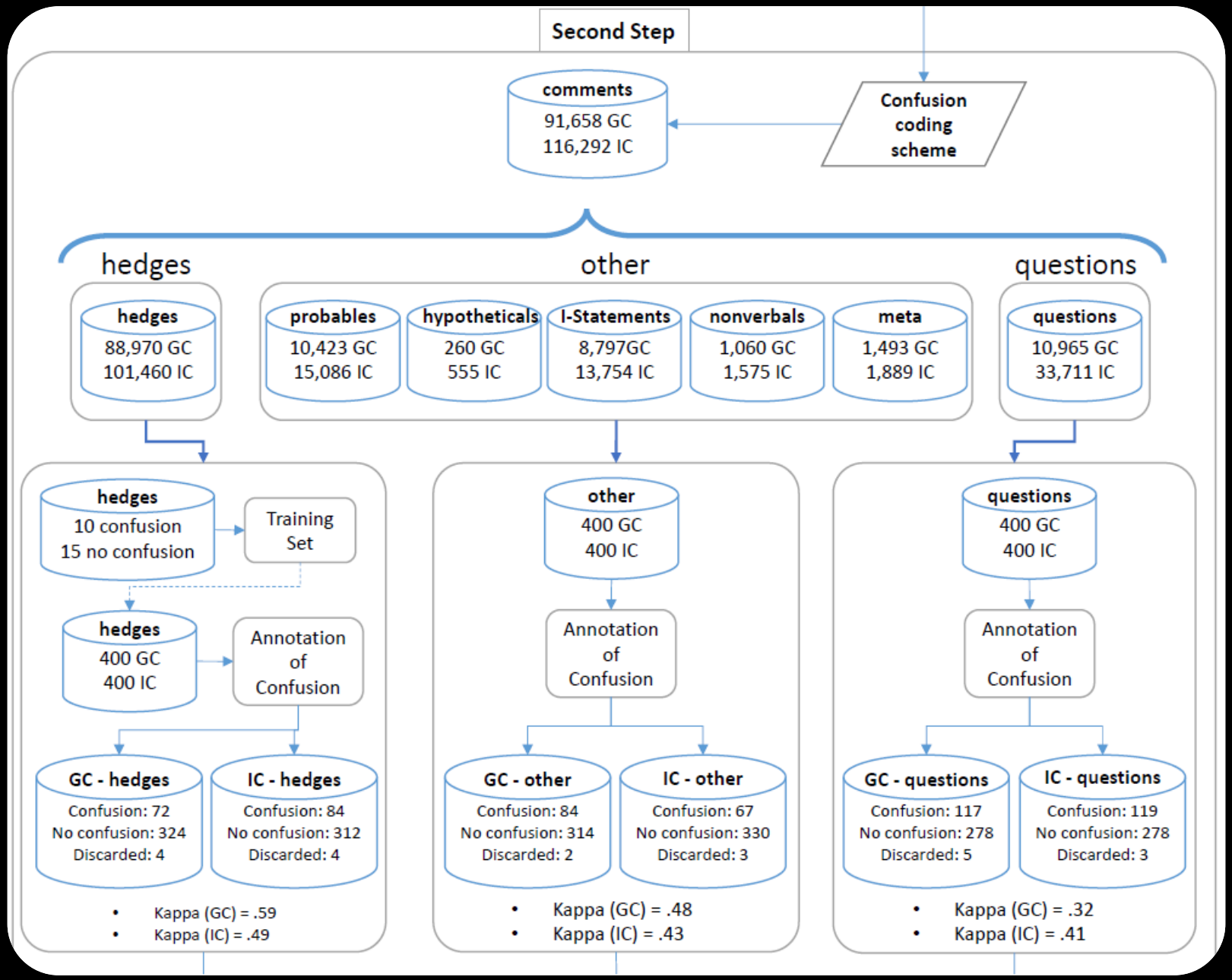


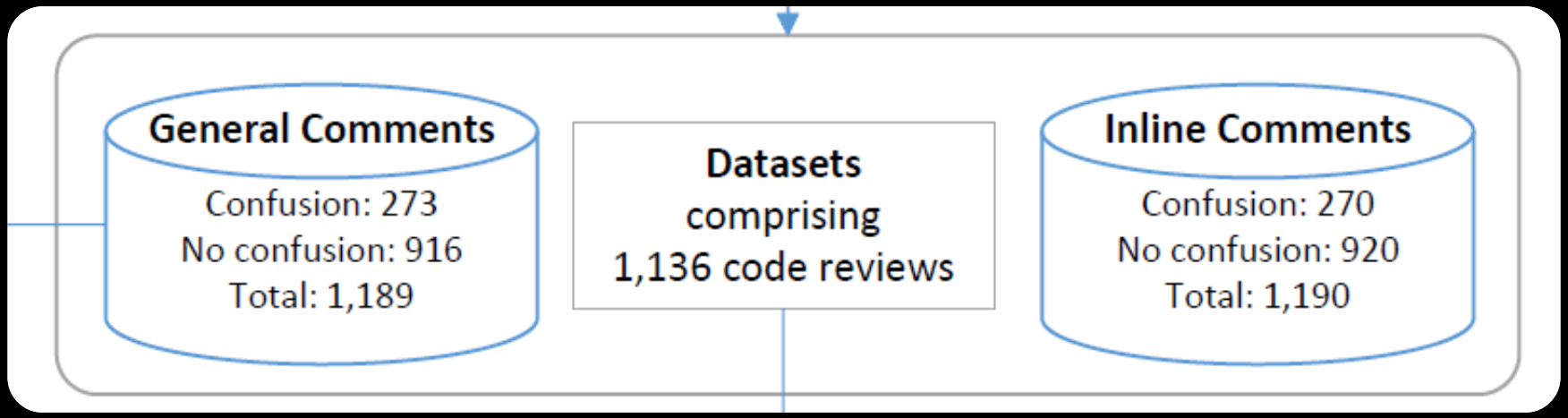
- GC – general comment
- IC – inline comment


Second Step
comments
91,658 GC
116,292 IC
Confusion coding scheme
hedges
other
questions
hedges
88,970 GC
101,460 IC
probables
10,423 GC
15,086 IC
hypotheticals
260 GC
555 IC
I-Statements
8,797GC
13,754 IC
nonverbals
1,060 GC
1,575 IC
meta
1,493 GC
1,889 IC
questions
10,965 GC
33,711 IC


Second Step
comments
91,658 GC
116,292 IC
Confusion coding scheme
hedges
other
questions
hedges
88,970 GC
101,460 IC
probables
10,423 GC
15,086 IC
hypotheticals
260 GC
555 IC
I-Statements
8,797GC
13,754 IC
nonverbals
1,060 GC
1,575 IC
meta
1,493 GC
1,889 IC
questions
10,965 GC
33,711 IC
hedges
10 confusion
15 no confusion
Training Set


Second Step
comments
91,658 GC
116,292 IC
Confusion coding scheme
hedges
other
questions
hedges
88,970 GC
101,460 IC
probables
10,423 GC
15,086 IC
hypotheticals
260 GC
555 IC
I-Statements
8,797GC
13,754 IC
nonverbals
1,060 GC
1,575 IC
meta
1,493 GC
1,889 IC
questions
10,965 GC
33,711 IC
hedges
10 confusion
15 no confusion
Training Set
hedges
400 GC
400 IC
Annotation of Confusion
GC - hedges
Confusion: 72
No confusion: 324
Discarded: 4
IC - hedges
Confusion: 84
No confusion: 312
Discarded: 4
Kappa (GC) = .59
Kappa (IC) = .49


Second Step
comments
91,658 GC
116,292 IC
Confusion coding scheme
hedges
other
questions
hedges
88,970 GC
101,460 IC
probables
10,423 GC
15,086 IC
hypotheticals
260 GC
555 IC
I-Statements
8,797GC
13,754 IC
nonverbals
1,060 GC
1,575 IC
meta
1,493 GC
1,889 IC
questions
10,965 GC
33,711 IC
hedges
10 confusion
15 no confusion
Training Set
hedges
400 GC
400 IC
Annotation of Confusion
GC - hedges
Confusion: 72
No confusion: 324
Discarded: 4
IC - hedges
Confusion: 84
No confusion: 312
Discarded: 4
Kappa (GC) = .59
Kappa (IC) = .49
other
400 GC
400 IC
Annotation of Confusion
GC - other
Confusion: 84
No confusion: 314
Discarded: 2
IC - other
Confusion: 67
No confusion: 330
Discarded: 3
Kappa (GC) = .48
Kappa (IC) = .43
questions
400 GC
400 IC
Annotation of Confusion
GC - questions
Confusion: 117
No confusion: 278
Discarded: 5
IC - questions
Confusion: 119
No confusion: 278
Discarded: 3
Kappa (GC) = .32
Kappa (IC) = .41


General Comments
Confusion: 273
No confusion: 916
Total: 1,189
Datasets
comprising
1,136 code reviews
Inline Comments
Confusion: 270
No confusion: 920
Total: 1,190

RQ1

GC - hedges
Confusion: 72
No confusion: 324
Discarded: 4

IC - hedges
Confusion: 84
No confusion: 312
Discarded: 4

- Kappa (GC) = .59
- Kappa (IC) = .49

GC - other
Confusion: 84
No confusion: 314
Discarded: 2

IC - other
Confusion: 67
No confusion: 330
Discarded: 3

- Kappa (GC) = .48
- Kappa (IC) = .43

GC - questions
Confusion: 117
No confusion: 278
Discarded: 5

IC - questions
Confusion: 119
No confusion: 278
Discarded: 3

- Kappa (GC) = .32
- Kappa (IC) = .41


12

Precision

		P	R	F
OneR	**GC**	**.875**	.194	.318
	IC	**.615**	.095	.165

Recall

		P	R	F
Multinomial Naive Bayes	**GC**	.209	**.944**	.342
	IC	.234	**.988**	.378

Precision and Recall

		P	R	F
JRip	**GC**	**.696**	**.542**	.609
Logistic	**IC**	**.434**	**.583**	.497

Precision

		P	R	F
OneR	**GC**	**.875**	.194	.318
	IC	**.615**	.095	.165

Recall

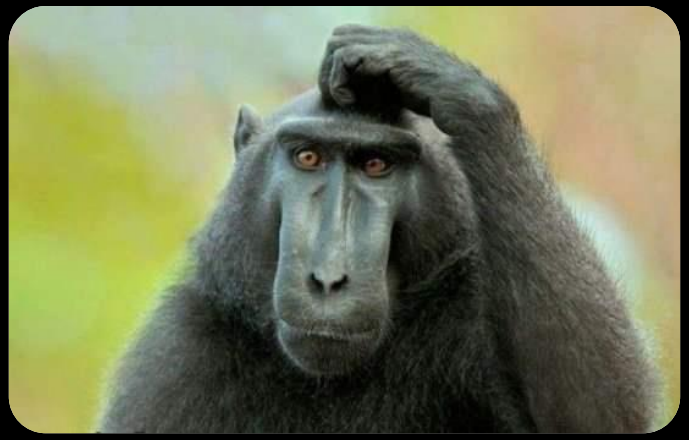
		P	R	F
Multinomial Naive Bayes	**GC**	.209	**.944**	.342
	IC	.234	**.988**	.378

Precision and Recall

		P	R	F
JRip	**GC**	**.696**	**.542**	.609
Logistic	**IC**	**.434**	**.583**	.497

Conclusions

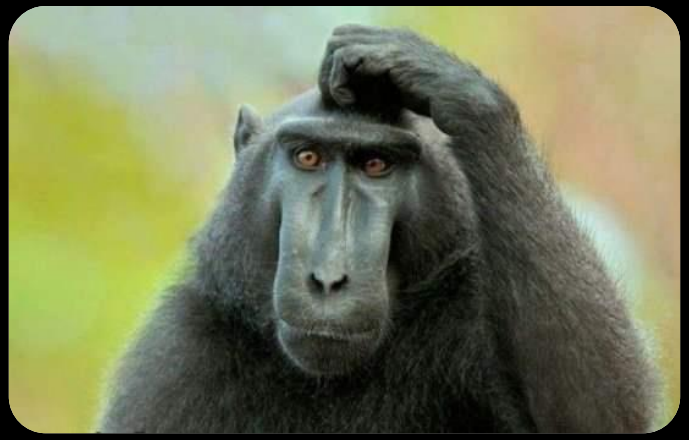
- Automatic detection of confusion:
  - Feasible task
  - Gold standard set



Conclusions

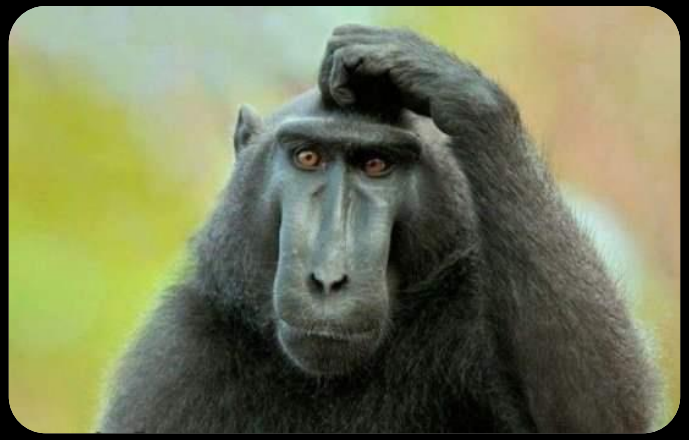
- Automatic detection of confusion:
  - Feasible task
  - Gold standard set

- Harder to identify confusion:
  - Inline comments



Conclusions

- Automatic detection of confusion:
  - Feasible task
  - Gold standard set
- Harder to identify confusion:
  - Inline comments
- “no-confusion” comments:
  - Suggestions
  - Politeness



Confusion in Context

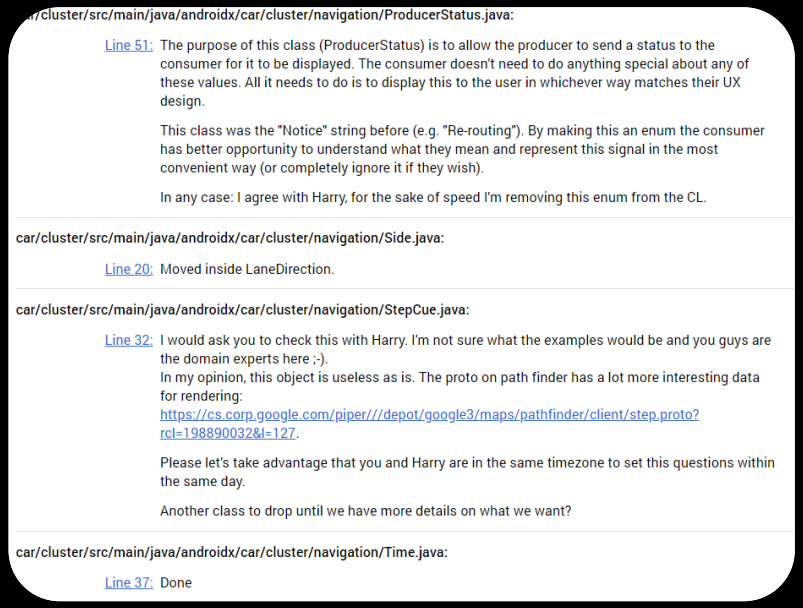
Reasons, impacts, and coping strategies

2nd Study

EBERT, F.; CASTOR, F.; NOVIELLI, N.; SEREBRENIK, A. Confusion in code reviews: Reasons, impacts and coping strategies. In: The 26th IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER'2019). Hangzhou, China: IEEE Computer Society, 2019.

Methodology

“what developers say”



car/cluster/src/main/java/androidx/car/cluster/navigation/ProducerStatus.java:

Line 51: The purpose of this class (ProducerStatus) is to allow the producer to send a status to the consumer for it to be displayed. The consumer doesn't need to do anything special about any of these values. All it needs to do is to display this to the user in whichever way matches their UX design.

This class was the "Notice" string before (e.g. "Re-routing"). By making this an enum the consumer has better opportunity to understand what they mean and represent this signal in the most convenient way (or completely ignore it if they wish).

In any case: I agree with Harry, for the sake of speed I'm removing this enum from the CL.

car/cluster/src/main/java/androidx/car/cluster/navigation/Side.java:

Line 20: Moved inside LaneDirection.

car/cluster/src/main/java/androidx/car/cluster/navigation/StepCue.java:

Line 32: I would ask you to check this with Harry. I'm not sure what the examples would be and you guys are the domain experts here ;-).
In my opinion, this object is useless as is. The proto on path finder has a lot more interesting data for rendering:
https://cs.corp.google.com/piper///depot/google3/maps/pathfinder/client/step.proto?rcl=198890032&l=127.

Please let's take advantage that you and Harry are in the same timezone to set this questions within the same day.

Another class to drop until we have more details on what we want?

car/cluster/src/main/java/androidx/car/cluster/navigation/Time.java:

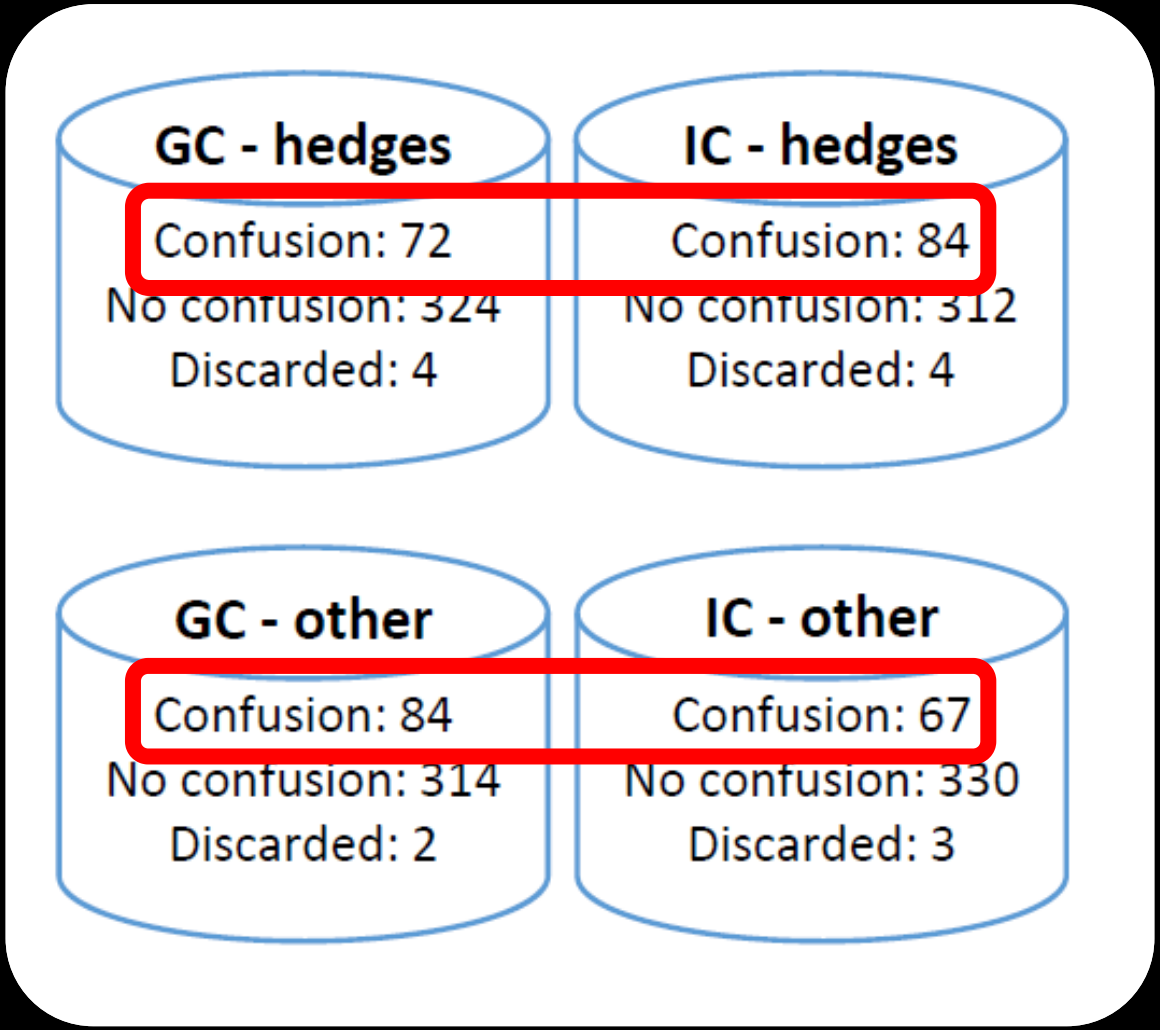
Line 37: Done

“what developers do”

Survey

- **How often** do you feel confused…?
- **What** usually makes you confused…?
- What is the **impact** of confusion…?
- What do you usually do to **overcome** confusion…?

Code Review Comments



Card sorting! Card sorting! Card sorting!

What Developers Say?

- 1st survey:
  - Android developers
  - 17 responses
  - Response rate: 0.45%

- 25 reasons
- 14 impacts
- 13 coping strategies

What Developers Say?

- 1st survey:
  - Android developers
  - 17 responses
  - Response rate: 0.45%

- 25 reasons
- 14 impacts
- 13 coping strategies

- 2nd survey:
  - Facebook & Twitter
  - 24 responses

- 25 reasons
- 16 impacts
- 14 coping strategies

What Developers Say?

- 1st survey:
  - Android developers
  - 17 responses
  - Response rate: 0.45%

- 25 reasons
- 14 impacts
- 13 coping strategies

- 2nd survey:
  - Facebook & Twitter
  - 24 responses

- 25 reasons
- 16 impacts
- 14 coping strategies

- 3rd survey:
  - Facebook & Twitter
  - 13 responses

- 25 reasons
- 16 impacts
- 14 coping strategies

What Developers Do?

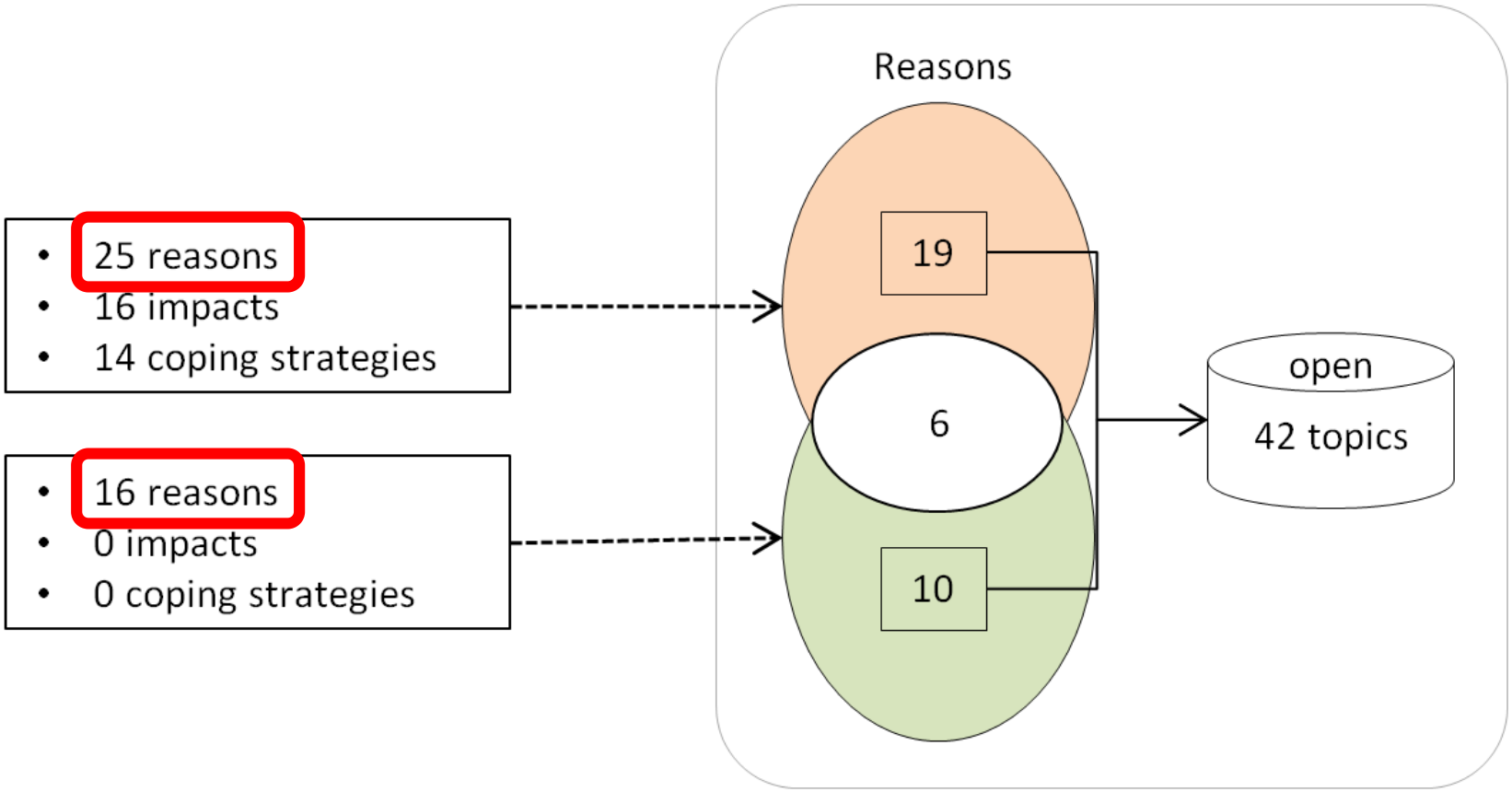
- 156 General Comments

  - 16 reasons
  - 0 impacts
  - 0 coping strategies

- 151 Inline Comments

  - 16 reasons
  - 0 impacts
  - 0 coping strategies

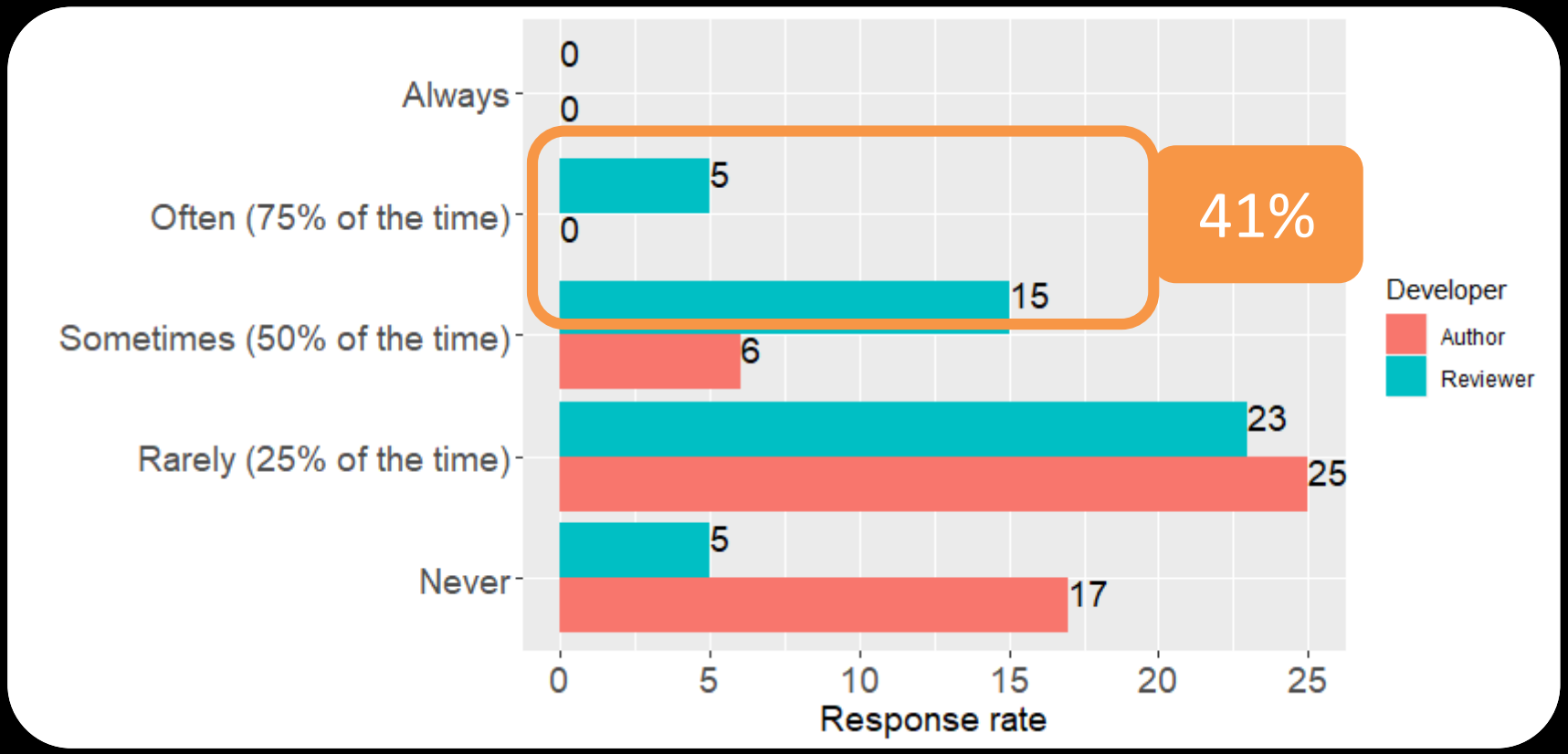
Triangulation Card Sorting



Confusion in Context Model Finalisation

- 30 reasons
- 14 impacts
- 13 coping strategies

Frequency of Confusion



	Reasons *30 topics (507)*	Impacts *14 topics (98)*	Coping strategies *13 topics (116)*
Review process *18 topics (120)*	Organisation of work (17) Issue tracker, version control (7) Unnecessary change (6) Not enough time (3) Dependency between changes (3) Code ownership (2) Community norms (2)	Delaying (31) Decreased review quality (11) Additional discussions (11) Blind approval (8) Review rejection (4) Increased development effort(4) Assignment to other reviewers (2)	Improved organisation of work (5) Delaying (2) Assignment to other reviewers (1) Blind approval (1)
Artifact *15 topics (300)*	Missing rationale (66) Discussion of the solution: non-func. (49) Unsure about system behavior (37) Lack of documentation (29) Discussion of the solution: strategy (29) Long, complex change (25) Lack of context (19) Discussion of the solution (14) Impact of change (11) Irreproducible bug (6) Lack of tests (5)	Better solution (1) Incorrect solution (1)	Small, clear changes (4) Improved documentation (4)
Developer *15 topics (124)*	Disagreement (18) Communicative intention (9) Language issues (3) Propagation of confusion (3) Fatigue (1) Noisy work environment (1)	Decreased confidence (10) Abandonment (6) Frustration (5) Propagation of confusion (2)	Information requests (36) Off-line discussions (12) Providing/accepting suggestions (10) Disagreement resolution (6)
Link *9 topics (177)*	Lack of familiarity with the existing code (47) Lack of programming skills (40) Lack of understanding of the problem (21) Lack of understanding of the change (17) Lack of familiarity with the technology (14) Lack of knowledge about the process (3)		Improved familiarity with the existing code (28) Testing the change (5) Improved familiarity with the technology (2)

	Reasons *30 topics (507)*	Impacts *14 topics (98)*	Coping strategies *13 topics (116)*
Review process *18 topics (120)*	Organisation of work (17) Issue tracker, version control (7) Unnecessary change (6) Not enough time (3) Dependency between changes (3) Code ownership (2) Community norms (2)	Delaying (31) Decreased review quality (11) Additional discussions (11) Blind approval (8) Review rejection (4) Increased development effort(4) Assignment to other reviewers (2)	Improved organisation of work (5) Delaying (2) Assignment to other reviewers (1) Blind approval (1)
Artifact *15 topics (300)*	Missing rationale (66) Discussion of the solution: non-func. (49) Unsure about system behavior (37) Lack of documentation (29) Di Lo La Di In Ir La	Better solution (1) Incorrect solution (1)	Small, clear changes (4) Improved documentation (4)
Developer *15 topics (124)*	Di Co La Pr Fa Noisy work environment (1)		sts (36) s (12) ng 0) lution (6)
Link *9 topics (177)*	Lack of familiarity with the existing code (47) Lack of programming skills (40) Lack of understanding of the problem (21) Lack of understanding of the change (17) Lack of familiarity with the technology (14) Lack of knowledge about the process (3)		Improved familiarity with the existing code (28) Testing the change (5) Improved familiarity with the technology (2)

2018 ACM/IEEE 11th International Workshop on Cooperative and Human Aspects of Software Engineering

The Structure of Software Design Discussions

Giovanni Viviani
University of British Columbia
vivianig@cs.ubc.ca

Calahan Janik-Jones
University of Toronto
cal.janik.jones@mail.utoronto.ca

Michalis Famelis
Université de Montréal
famelis@iro.umontreal.ca

Gail C. Murphy
University of British Columbia
murphy@cs.ubc.ca

	Reasons *30 topics (507)*	Impacts *14 topics (98)*	Coping strategies *13 topics (116)*
Review process *18 topics (120)*	Organisation of work (17) Issue tracker, version control (7) Unnecessary change (6) Not enough time (3) Dependency between chang Code ownership (2) Community norms (2)	Delaying (31) Decreased review quality (11)) ort(4) wers (2)	Improved organisation of work (5) Delaying (2) Assignment to other reviewers (1) Blind approval (1)
Artifact *15 topics (300)*	Missing rationale (66) Discussion of the solution: Unsure about system behav Lack of documentation (29) Discussion of the solution: Long, complex change (25) Lack of context (19) Discussion of the solution (Impact of change (11) Irreproducible bug (6) Lac		Small, clear changes (4) Improved documentation (4)
Developer *15 topics (124)*	Dis Co Lar Pro Fat Noi		ts (36) (12) g) ution (6)
Link *9 topics (177)*	Lac Lac Lac Lac Lac Lac		y with le (28) (5) y with (2)

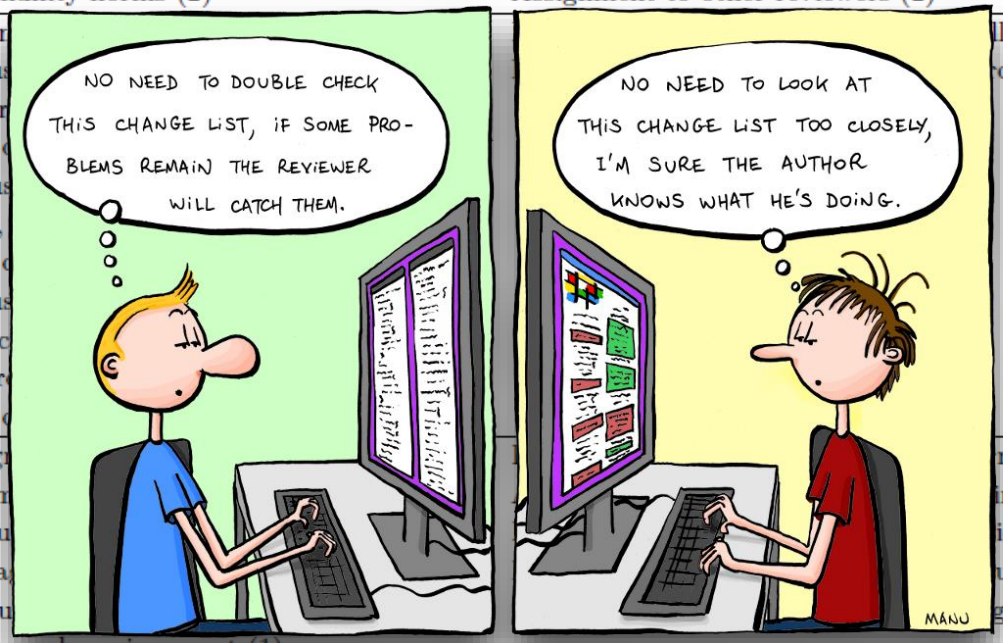
Empir Software Eng (2018) 23:835–904
https://doi.org/10.1007/s10664-017-9548-7

CrossMark

The impact of rapid release cycles on the integration delay of fixed issues

Daniel Alencar da Costa[1] · Shane McIntosh[2] · Christoph Treude[3] · Uirá Kulesza[4] · Ahmed E. Hassan[5]

	Reasons *30 topics (507)*	Impacts *14 topics (98)*	Coping strategies *13 topics (116)*
Review process *18 topics (120)*	Organisation of work (17) Issue tracker, version control (7) Unnecessary change (6) Not enough time (3) Dependency between changes (3) Code ownership (2) Community norms (2)	Delaying (31) Decreased review quality (11) Additional discussions (11) Blind approval (8) Review rejection (4) Increased development effort(4) Assignment to other reviewers (2)	Improved organisation of work (5) Delaying (2) Assignment to other reviewers (1) Blind approval (1)
Artifact *15 topics (300)*	Missin Discus Unsur Lack Discus Long, Lack Discus Impac Irrepr Lack		ll, clear changes (4) roved documentation (4)
Developer *15 topics (124)*	Disag Comn Langu Propa Fatigu Noisy work environment (1)		mation requests (36) ine discussions (12) iding/accepting uggestions (10) greement resolution (6)
Link *9 topics (177)*	Lack of familiarity with the existing code (47) Lack of programming skills (40) Lack of understanding of the problem (21) Lack of understanding of the change (17) Lack of familiarity with the technology (14) Lack of knowledge about the process (3)		Improved familiarity with the existing code (28) Testing the change (5) Improved familiarity with the technology (2)



2014 IEEE International Conference on Software Maintenance and Evolution

Writing Acceptable Patches: An Empirical Study of Open Source Project Patches

Yida Tao*, DongGyun Han† and Sunghun Kim*
*Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
{idagoo, hunkim}@cse.ust.hk
†KAIST Institute for IT Convergence
Korea Advanced Institute of Science and Technology
handk@kaist.ac.kr

			Coping strategies *13 topics (116)*
		w quality (11)	Improved organisation of work (5)
		ssions (11)	Delaying (2)
		(8)	Assignment to other reviewers (1)
		n (4)	Blind approval (1)
		opment effort(4)	
		other reviewers (2)	
	Missing rationale (56)	Better solution (1)	Small, clear changes (4)
	Discussion of the solution: non-func. (49)		
	Unsure about system behavior (37)		
	Lack of documentation (29)		
Artifact *15 topics (300)*	Discussion of the solution: strategy (29)		
	Long, complex change (25)		
	Lack of context (19)		
	Discussion of the solution (14)		
	Impact of change (11)		
	Irreproducible bug (6)		
	Lack of tests (5)		
	Disagreement (18)	Decreased confidence (10)	Information requests (36)
			Off-line discussions (12)
			Providing/accepting suggestions (10)
			Disagreement resolution (6)
			Improved familiarity with the existing code (28)
			Testing the change (5)
			Improved familiarity with the technology (2)

2015 IEEE/ACM 37th IEEE International Conference on Software Engineering

Helping Developers Help Themselves: Automatic Decomposition of Code Review Changesets

Mike Barnett
Microsoft Research
Redmond, WA, USA
mbarnett@microsoft.com

Christian Bird
Microsoft Research
Redmond, WA, USA
cbird@microsoft.com

João Brunet
Federal University of Campina Grande
Campina Grande, Paraíba, Brazil
joao.arthur@computacao.ufcg.edu.br

Shuvendu K. Lahiri
Microsoft Research
Redmond, WA, USA
shuvendu@microsoft.com

2015 12th Working Conference on Mining Software Repositories

Partitioning Composite Code Changes to Facilitate Code Review

Yida Tao and Sunghun Kim
The Hong Kong University of Science and Technology
Department of Computer Science and Engineering
{idagoo, hunkim}@cse.ust.hk

	Reasons *30 topics (507)*	Impacts *14 topics (98)*	Coping strategies *13 topics (116)*
Review process *18 topics (120)*	Organisation of work (17) Issue tracker, version control (7) Unnecessary change (6) Not enough time (3) Dependency between changes (3) Code ownership (2) Community norms (2)	Delaying (31) Decreased review quality (11) Additional discussions (11) Blind approval (8) Review rejection (4) Increased development effort(4) Assignment to other reviewers (2)	Improved organisation of work (5) Delaying (2) Assignment to other reviewers (1) Blind approval (1)
Artifact *15 topics (300)*	Missing rationale (66) Discussion of the solution: non-func. (49) Unsure about system behavior (37) Lack of documentation (29) Discussion of the solution: strategy (29) Long, complex change (25) Lack of context (19) Discussion of the solution (14) Impact of change (11) Irreproducible bug (6) Lack of tests (5)	Better solution (1) Incorrect solution (1)	Small, clear changes (4) Improved documentation (4)
Developer *15 topics (124)*	Disagreement (18) Communicative intention (9) Language issues (3) Propagation of confusion (3) Fatigue (1) Noisy work environment (1)	Decreased confidence (10) Abandonment (6) Frustration (5) Propagation of confusion (2)	Information requests (36) Off-line discussions (12) Providing/accepting suggestions (10) Disagreement resolution (6)
Link *9 topics (177)*	Lack of familiarity with the existing code (47) Lack of programming skills (40) Lack of understanding of the problem (21) Lack of understanding of the change (17) Lack of familiarity with the technology (14) Lack of knowledge about the process (3)		Improved familiarity with the existing code (28) Testing the change (5) Improved familiarity with the technology (2)

Conclusions

- Confusion is present!
  - “Developers said!” (survey)
  - “Developers did!” (code review comments)

Conclusions

- Confusion is present!
  - “Developers said!” (survey)
  - “Developers did!” (code review comments)

- Confusion in context:
  - 30 reasons
  - 14 impacts
  - 13 coping strategies

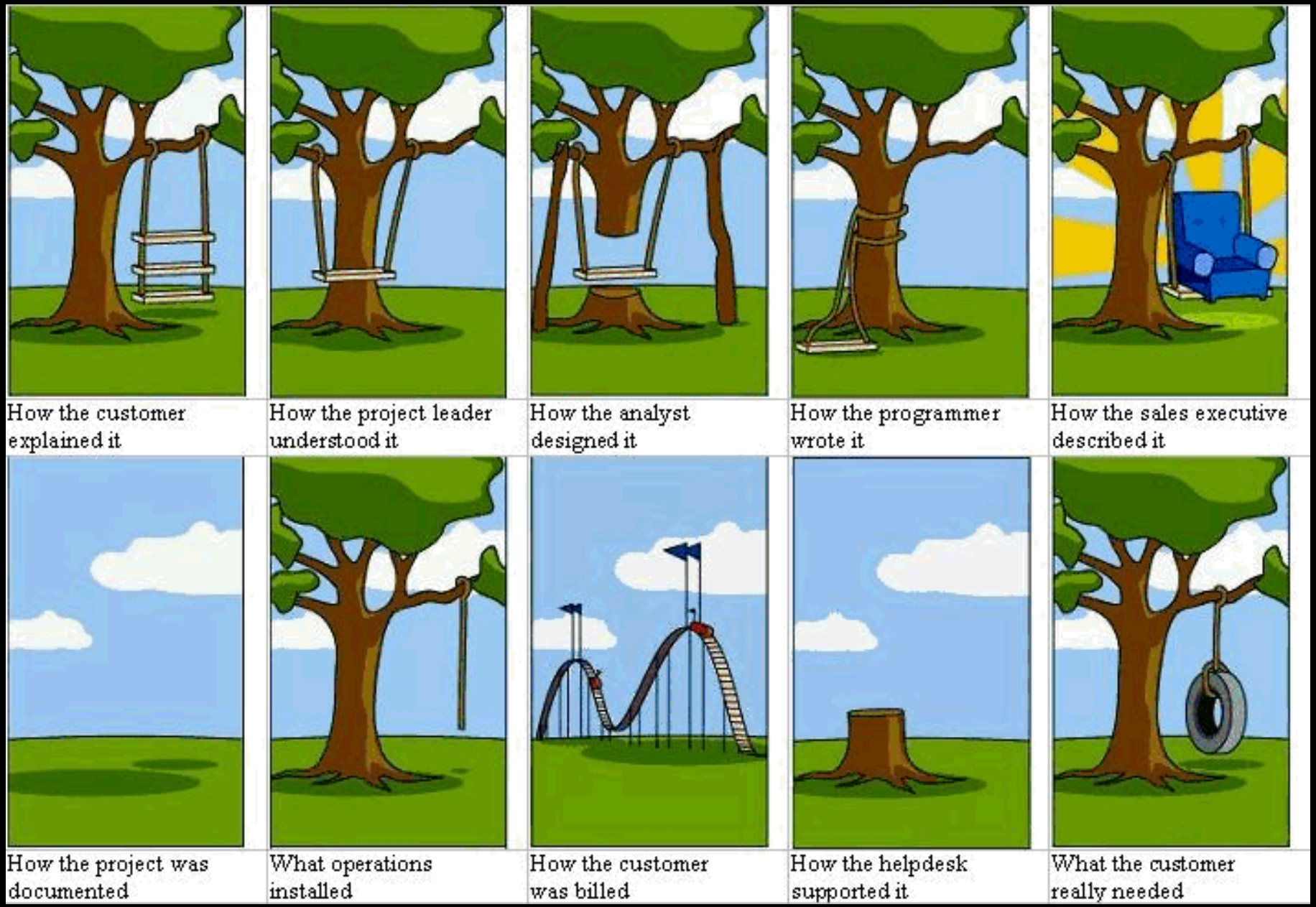
Conclusions

- Confusion is present!
  - “Developers said!” (survey)
  - “Developers did!” (code review comments)
- Confusion in context:
  - 30 reasons
  - 14 impacts
  - 13 coping strategies
- Topics not studies yet!

Communicative Intentions of Questions

3rd Study

EBERT, F.; CASTOR, F.; NOVIELLI, N.; SEREBRENIK, A. Communicative intention in code review questions. In: The 34th IEEE International Conference on Software Maintenance and Evolution (ICSME). Madrid, Spain: IEEE Computer Society, 2018. p. 519–523. ISSN 2576-3148.

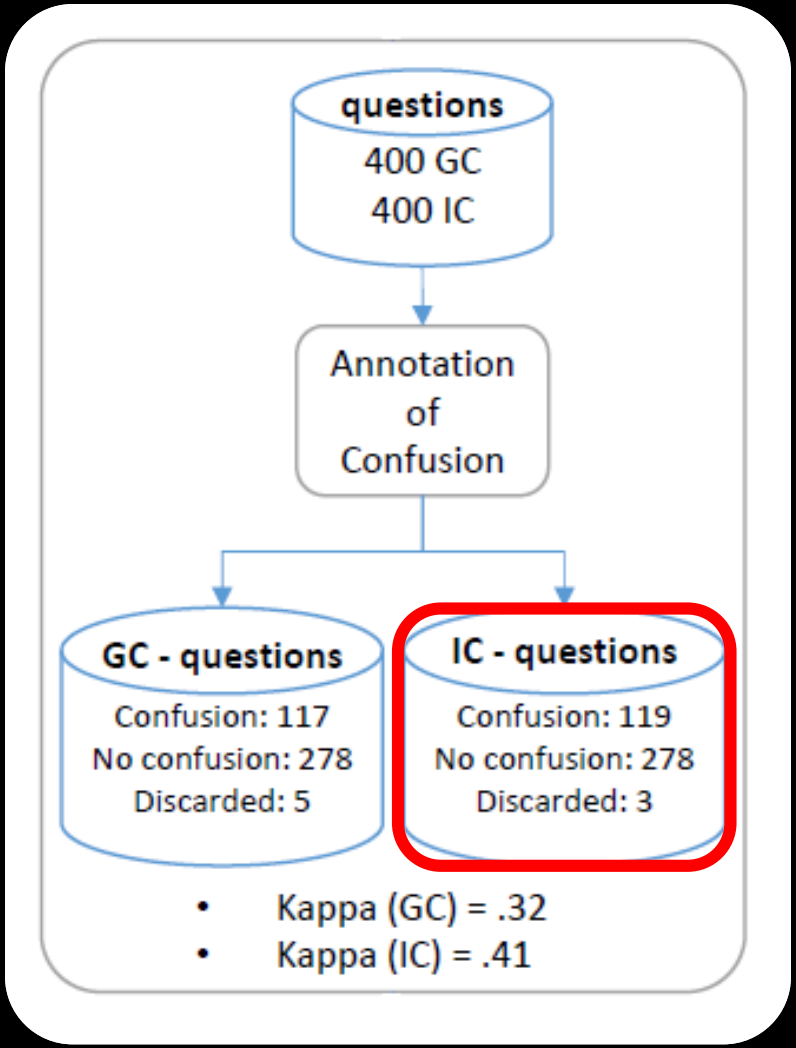

How the customer explained it
How the project leader understood it
How the analyst designed it
How the programmer wrote it
How the sales executive described it
How the project was documented
What operations installed
How the customer was billed
How the helpdesk supported it
What the customer really needed

What do you mean when you ask a question?

Communicative Intentions of Questions

- RQ1: How frequent are questions in code reviews?
- RQ2: What are the communicative intentions expressed in the developers' questions in code reviews?

Dataset


questions
400 GC
400 IC
Annotation of Confusion
GC - questions
Confusion: 117
No confusion: 278
Discarded: 5
IC - questions
Confusion: 119
No confusion: 278
Discarded: 3
• Kappa (GC) = .32
• Kappa (IC) = .41

Exploratory Case Study

- First step:
  - 25 comments -> 49 questions
  - 11 categories

Exploratory Case Study

- First step:
  - 25 comments -> 49 questions
  - 11 categories

- Second step:
  - 400 comments -> 499 questions
  - 12 categories

How frequent are questions in code reviews?

	Number of questions
General comments	12,686 (25%)
Inline comments	**37,712 (75%)**
Total	50,398

How frequent are questions in code reviews?

	Number of questions
General comments	12,686 (25%)
Inline comments	**37,712 (75%)**
Total	50,398

	General comments	Inline comments
With at lest one Question	10,965 (1,65%)	**33,711 (14,50%)**
Without any Questions	649,880 (98%)	198,760 (85%)
Total	660,845	232,471

Soliciting an action

Suggestion

"Maybe introduce an additional line between 'abc' and 'def'?"

Request for action

"Can you make these different?"

Information seeking

Information	“When can this be null?”
Confirmation	“Shouldn’t this just be a failure?”
Rationale	“Why is this included?”
Clarification	"Can you clarify what you mean?"
Opinion	“Which name do you suggest?”

Attitudes and Emotions

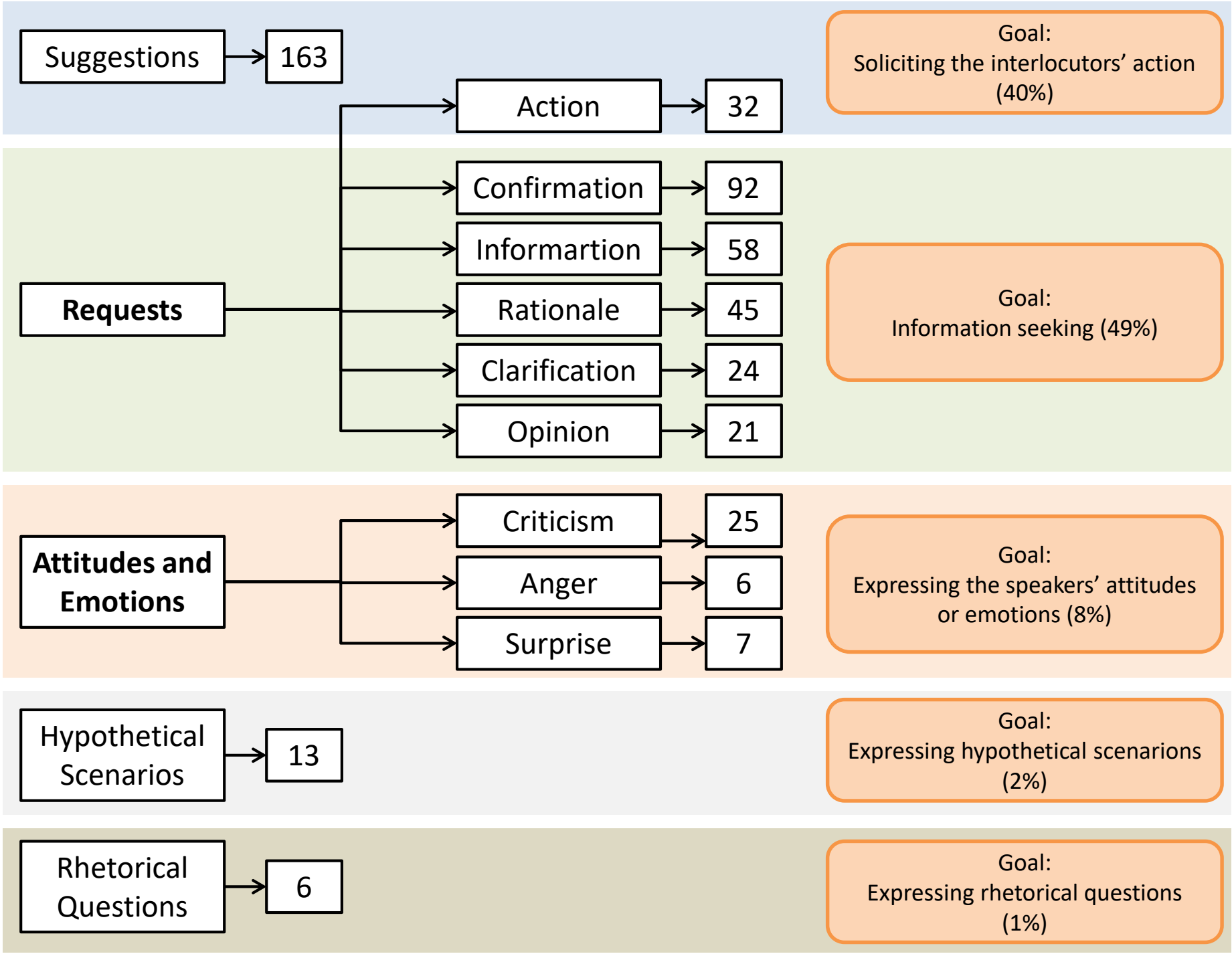
Criticism	"Do you really want to return the address of a local variable here?"
Anger	"wtf? you really want reflection here?"
Surprise	"Is this true? That seems mildly surprising"

Hypothetical scenarios

"What about if an already Jack server is running?"

Rhetorical questions

"Isn't the case that you illustrated (0.9ms being decremented as 0) applicable in both solutions? Yes"


Suggestions
163
Action
32
Goal:
Soliciting the interlocutors' action (40%)
Requests
Confirmation
92
Informartion
58
Rationale
45
Clarification
24
Opinion
21
Goal:
Information seeking (49%)
Attitudes and Emotions
Criticism
25
Anger
6
Surprise
7
Goal:
Expressing the speakers' attitudes or emotions (8%)
Hypothetical Scenarios
13
Goal:
Expressing hypothetical scenarions (2%)
Rhetorical Questions
6
Goal:
Expressing rhetorical questions (1%)

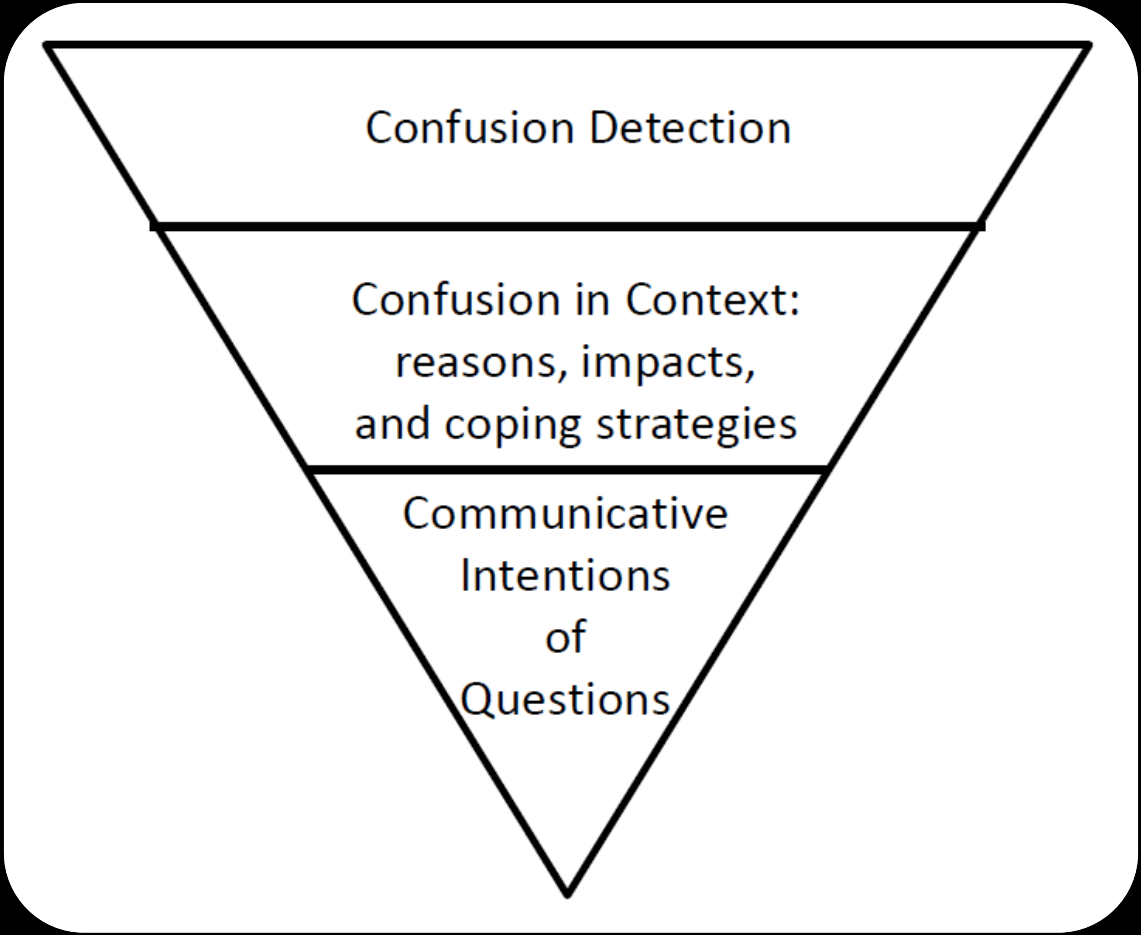
Conclusions

- Questions are more present in the IC than GC

Conclusions

- Questions are more present in the IC than GC
- Questions:
  - Not only information seeking
  - Suggestions
  - Attitude and emotions

Understanding Confusion in Code Reviews



Felipe Ebert – f.ebert@tue.nl