Code Reviews

Felipe Ebert f.ebert@tue.nl

Software Engineering and Technology (SET)





Code Review

2



Code Inspections

- It is a formal structured process to find defects
- Format:
 - Highly structured process
 - Based on line-by-line group reviews
 - Done in extended meetings

- Roles:
 - Moderator
 - Designer
 - Coder
 - Tester

Modern Code Review

- Informal
- Tool-based
- Occurs regularly in practice

0=Math.abs(0)1 while (0:=0)(if (A>0) A=A=01 else 0=0=A1

A+Mathiabs(A)1

integer

026

> return AS

idAlgorithm Cint A. int

BACCHELLI, A.; BIRD, C. Expectations, outcomes, and challenges of modern code review. ICSE, USA, IEEE Press, 2013.







Code Improvement Alternative Solutions



Find Defects



Code Improvement Alternative Solutions





Find Defects





Code Improvement Alternative Solutions



A Case Study



COHEN, J.; TELEKI, S.; BROWN, E. Best Kept Secrets of Peer Code Review. Massachusetts, EUA: Smart Bear Inc., 2006.

A Case Study





COHEN, J.; TELEKI, S.; BROWN, E. Best Kept Secrets of Peer Code Review. Massachusetts, EUA: Smart Bear Inc., 2006.

Who Is Doing Code Reviews?



Main repository

Review-then-commit (pre-commit) Commit-then-review (post-commit)



www.menti.com 1969 8585

Main repository

Review-then-commit (pre-commit)

Commit-then-review (post-commit)



• Review-then-commit

- Pros:
 - Quality standard are met
 - Review has been performed
 - Team won't be affected by bugs



• Review-then-commit

- Pros:
 - Quality standard are met
 - Review has been performed
 - Team won't be affected by bugs
- Cons:
 - Decreases productivity



• Review-then-commit

• Pros:

- Quality standard are met
- Review has been performed
- Team won't be affected by bugs

• Cons:

Decreases productivity

Code change is merged

Commit-then-review

• Pros:

- Commit changes continuously
- Members see the code changes

Review-then-commit

• Pros:

- Quality standard are met
- Review has been performed
- Team won't be affected by bugs

• Cons:

Decreases productivity

Code change is merged

Commit-then-review

• Pros:

- Commit changes continuously
- Members see the code changes

• Cons:

- Poor code might make into repository
- Switching back to fix the bug

- Email pass-around
 - Reviewers receive code changes sent by the source code management system via e-mail

- Email pass-around
 - Reviewers receive code changes sent by the source code management system via e-mail
- Over the shoulder
 - The code's author gives a walkthrough to a colleague

- Email pass-around
 - Reviewers receive code changes sent by the source code management system via e-mail
- Over the shoulder
 - The code's author gives a walkthrough to a colleague
- Pair programming
 - Two authors develop the code together at the same workstation

- Email pass-around
 - Reviewers receive code changes sent by the source code management system via e-mail
- Over the shoulder
 - The code's author gives a walkthrough to a colleague
- Pair programming
 - Two authors develop the code together at the same workstation
- Tool assisted







General Comment

Uploaded patch set 2.	Patchset 2	May 07, 2015 7:33 PM 🗸
1 comment	Patchset 1	May 07, 2015 7:36 PM 🗸
Code-Review +2	Patchset 2	May 07, 2015 7:37 PM 🗸
Added to reviewer:		May 07, 2015 7:45 PM 🗸
	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 ifndefARM		
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));</uint32_t>		
fini_array_count_ = static_cast <uint32_t>(d->d_un.d_val) / sizeof(ElfW(Addr));</uint32_t>		
Snippets from linker/linker.cpp		

Inline Comment

149682 Properly align init/fini_arrays for crtend.o 🖉 libc/arch-common/bionic/crtend.S 🤝	File 3 of 4 <u>Prev</u> <u>Up</u> <u>Next</u>
Base \neg gitiles \rightarrow Patchset 1 \neg gitiles DOWNLOAD \neg	SHOW BLAME 💉 Diff view: 📘 🖨 🌣
<pre>Base * guiles = Paidiset * guiles DOWNLOAD * 30 31 > .section .preinit_array, "aw" 32 > ASM_PTR_SIZE(0) 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) </pre>	Bit View. Diff View.
	j0 #endif







Crucible

GitHub





Code Review Tools



What to Look For in a Code Review?



Code Reviews Are Not Free!!

34



Existing Challenges...

- Receiving feedback in a timely manner
- Review size
- Understanding the code change, its purpose, its context
- Obtaining insightful feedback
- Nitpicking
- Finding relevant information
- Identifying an appropriate reviewer

Existing Challenges...

- Receiving feedback in a timely manner
- Review size
- Understanding the code change, its purpose, its context
- Obtaining insightful feedback
- Nitpicking
- Finding relevant information
- Identifying an appropriate reviewer

Why confusion?



Why confusion?

 "we found that context and change understanding is the key of any review"



Why confusion?

- "we found that context and change understanding is the key of any review"
- Delays, bugs, frustration



Confusion is...

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

Ebert, Castor, Novielli, Serebrenik, "Confusion detection in code reviews" ICSME, 549–553, 2017 Jordan et al., "Expressing uncertainty in computer-mediated discourse: Language as a marker of intellectual work," Discourse Processes, 660–692, 2012.

Examples of Confusion





Research Goals

• Can we **identify** manually and automatically confusion in code review discussions?

• What are the **reasons** for confusion, its **impacts**, and how developers are **dealing** with it?

• Android:

- Code reviews: 1,136
 - General comments: 1,200
 - Inline comments: 1,200



If this is true, maxX is less than minX? That seems weird and confusing.

confusion

Could you use getString(R.string.unknown) to ensure this is translated?

no-confusion



Ebert, Castor, Novielli, Serebrenik, "Confusion detection in code reviews" ICSME, 549–553, 2017

Confusion Effects in Code Reviews

• What are the **reasons** for confusion?

• What are the **impacts** of confusion?

• What are the **coping strategies** developers use to deal with it?

Methodology

SURVEY Very Satisfied Satisfied Neutral Unsatisfied Very Unsatisfied

"What developers say"



Card Sorting! Card Sorting! Card Sorting!













41% feel confused at least 50% of the time



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

Reasons

	Organisation of work (17)
Review	Issue tracker, version control (7)
DROGOSS	Unnecessary change (6)
18 topics	Net enough time (2)
10 topics (100)	Not enough time (3)
(120)	Cada arranges (3)
	Code ownership (2)
	Community norms (2)
	Missing rationale (66)
	Discussion of the solution: non-func. (49)
	Unsure about system behavior (37)
	Lack of documentation (29)
Artifact	Discussion of the solution: strategy (29)
15 topics	Long, complex change (25)
(300)	Lack of context (19)
	Discussion of the solution: correctness (14)
	Impact of change (11)
	Irreproducible bug (6)
	Lack of tests (5)
	Disagreement (18)
Develope	erCommunicative intention (9)
15 topics	Language issues (3)
(124)	Propagation of confusion (3)
	Fatigue (1)
	Noisy work environment (1)
	Lack of familiarity with the existing code (47)
Link	Lack of programming skills (40)
9 topics	Lack of understanding of the problem (21)
(177)	Lack of understanding of the change (17)
	Lack of familiarity with the technology (14)
	Lack of knowledge about the process (3)

Reason: Missing Rationale



Reason: Missing Rationale



Ebert, "From Transient Information to Persistent Documentation: Enhancing Software Documentation", ICSME, 849-853, 2020

	Impacts
	Delaying (31)
Review	Decreased review quality (11)
process	Additional discussions (11)
18 topics	Blind approval (8)
(120)	Review rejection (4)
	Increased development $effort(4)$
	Assignment to other reviewers (2)
	Better solution (1)
	Incorrect solution (1)
(/	
	Decreased confidence (10)
Develope	Abandonment (6)
15 topics	Frustration (5)
(124)	Propagation of confusion (2)
Link	
Link 9 topics	

Coping Strategies Improved organisation Review of work (5) Delaying (2) process 18 topics Assignment to (120)other reviewers (1) Blind approval (1) Small, clear changes (4) Improved documentation (4) Artifact 15 topics (300)Information requests (36) Developer Off-line discussions (12) Providing/accepting 15 topics (124)suggestions (10) Disagreement resolution (6) Improved familiarity with Link the existing code (28)9 topics Testing the change (5)(177) Improved familiarity with the technology (2)

Code Reviews & Confusion





Interested in working with code reviews? Ping me! =D

Felipe Ebert - f.ebert@tue.nl