

Code Review Quality: How Developers See It

Oleksii Kononenko

Olga Baysal

Michael W. Godfrey

88人のMozilla開発者にコードレビューについての認識を
インタビュー（質的な調査）してまとめた内容

背景と目的

- ・ コードレビューはソフトウェアの品質を保つのに必要な行程
 - 時間, 労力はかかるが, 大規模・長命なプロジェクトで取り入れられている
- ・ レビューに関して盛んに研究されているが, 開発者の認識はわからない
- ・ 量的な先行研究に対して, 開発者にインタビューをし, 質的に調査をした

リサーチクエスチョン

RQ1

- ・ レビューの実施方法（Mozilla開発者）

RQ2

- ・ レビューの時間に影響する要因
- ・ レビューの採否に影響する要因

RQ3

- ・ コードレビューの質を評価するのに使う要因
 - ・ RQ3.1：パッチの質の評価方法
 - ・ RQ3.2：良いレビューの定義
 - ・ RQ3.3：レビューの質に関係する要因

RQ4

- ・ 開発者がレビュー時に直面する課題

分析対象

- ・ Mozillaプロジェクトの開発者
 - Mozillaプロジェクトは, 有名・大規模・長命
- ・ Bugzillaからメールアドレスを抽出 (3,142件)
- ・ アクティブな開発者403人にメールを送信
 - 15以上のパッチがプロジェクトに承認されている
 - 6ヶ月以上活動している開発者
- ・ **返信があった88人 (22%) の回答を分析**
 - 質問は5-10分でできる内容

質問内容と方法

- ・ 開発者自身に対する質問
- ・ レビューに対する認識
 - **リッカート尺度**
 - ・ 同意する or しない を5段階で回答してもらう

全く同意しない 同意しない どちらでもない 同意する 強く同意する



- **自由記述式**
 - ・ 質問に対する答えの根拠
 - ・ 開発者がレビューで感じている課題など

自由記述式回答の分析方法

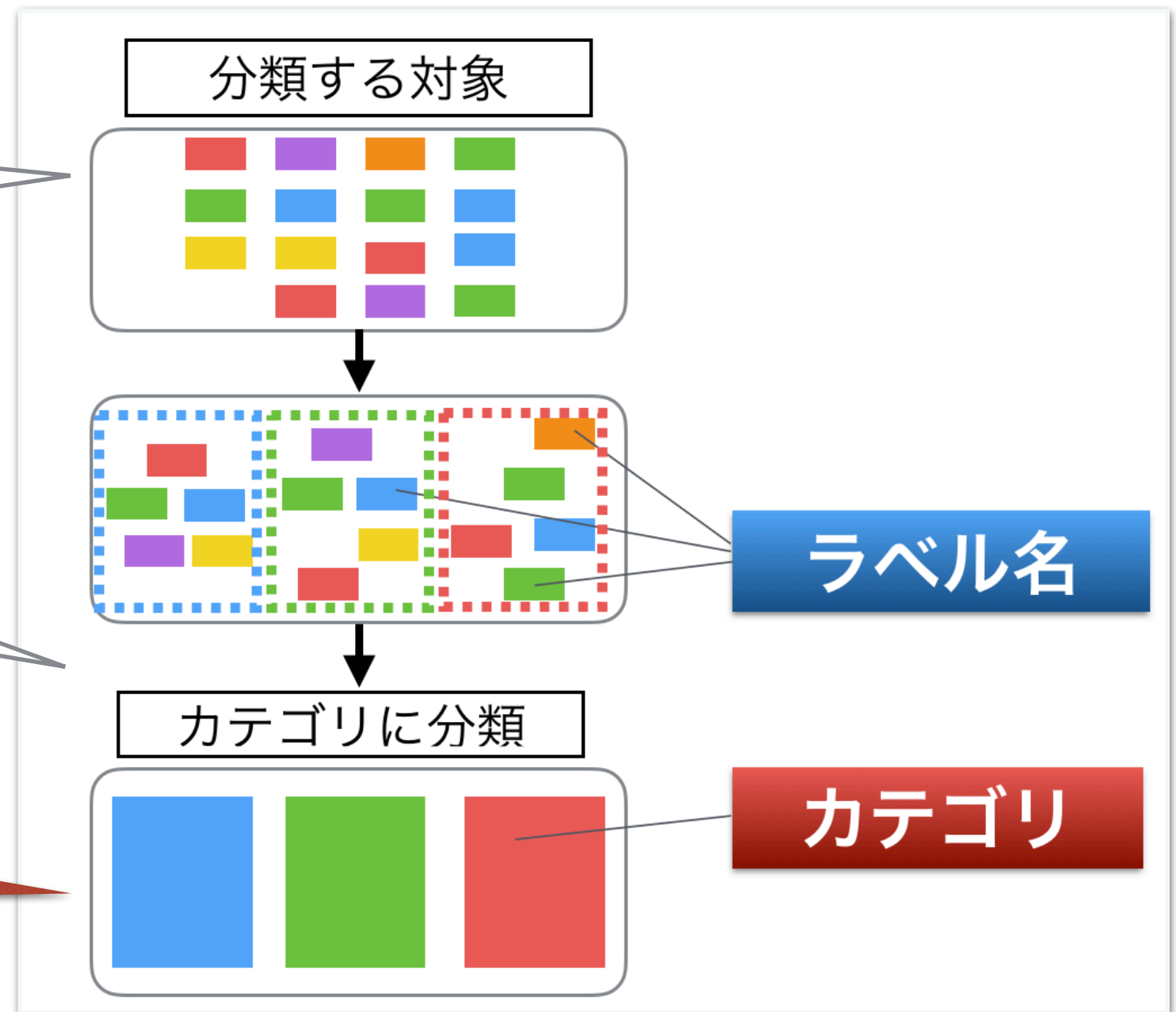
・ グランデッドセオリー

一連の作業を**オープン・コーディング**という
(M. Miles and A. Huberman. Qualitative Data Analysis:
An Expanded Sourcebook. SAGE Publications, 1994.)

Kononenkoが
938種類の”card”に分類

KononenkoとBaysalが
ラベル名 (theme) を作成
さらにカテゴリに分類

最終的に
30種類のカテゴリに分類



結果：RQ1, レビューの実施方法

- ・ **作業量（一週間）**

- パッチ提出またはレビュー
 - ・ 5パッチ以下：提出 69%, レビュー 57%
 - ・ 10パッチ以上活動する開発者は片方の作業に専念している傾向がある

- ・ **環境**

- レビューはほとんどBugzillaでやる, レビューツールはあまり使わない
- 議論はIRC (Internet Relay Chat) などを使う

結果：RQ2, 開発者が考えるレビューの時間に影響する要因

質：開発者の意見
量：先行研究

質：パッチサイズ：◎

量：小さいサイズのコミットは応答が速い

質：経験：◎

量：経験が多いとレビューの応答が速い

質：緊急性, 優先度：△

量：バグトリアージではよく使われるけど

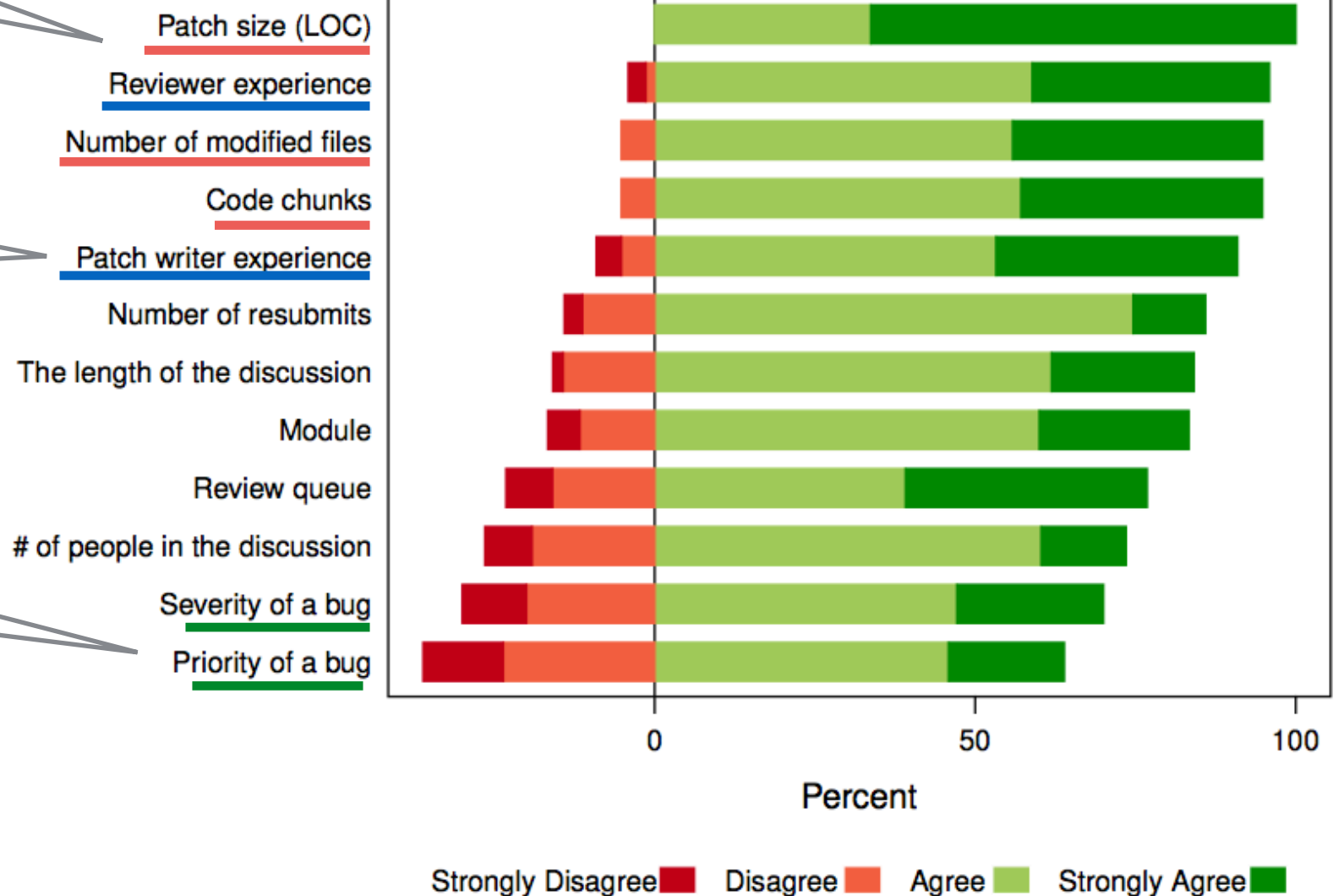


Figure 1: Factors influencing code review time.

先行研究の結果と開発者の認識はだいたい同じ

結果：RQ2, 開発者が考えるレビューの**時間**に影響する要因

Category	Q13	
	#Q	#R
Code quality	24%	30%
Testing	6%	9%
Time constraints	—	—
Change scope/rationale	9%	12%
Understanding code change/base	6%	9%
Human factors	—	—
Tools	4%	5%
Communication	2%	2%
Change complexity	18%	23%
Relationship/trust	3%	5%
Usefulness	—	—
Workload	—	—
Submitter related	3%	5%
Architecture/design	—	—
Reviewer related	—	—
Discussion	—	—
Conformance to project goals	—	—
Bug type	9%	12%
Selecting correct reviewer	7%	12%
Performance	—	—
Integration into code base	—	—
Security	—	—
Memory management	—	—
Familiarity with the author	—	—
Thorough feedback	—	—
Catching bugs	—	—
Organizational factors	—	—
Documentation	—	—
Context switch	—	—
Irrelevant	9%	12%
Total:	67	43
Average percent agreement	96.6%	

コードの質：◎

可読性, 変数名 などからコードの質が想像できる
“hunk”

変更範囲/割合：○

小さいバグの方が大きいバグよりすぐ確認できる

バグタイプ：○（意外, surprisingly）

バグの原因が曖昧だとレビューに時間がかかる

Notes: #Q: the number of quotes, #R: the number of respondents, Q11: factors affecting decision, Q13: factors affecting time, Q14: patch quality, Q15: characteristics of code review quality, Q17: other factors affecting review quality, Q18: challenges.

結果：RQ2, 開発者が考えるレビューの採否に影響する要因

先行研究の結果と開発者の認識は違うところがある

経験：◎

緊急性, 優先度：○（意外）

パッチサイズ：×

レビュー負荷：××

量：レビューキューが小さい開発者はパッチを拒否しやすい

質：開発者の意見
量：先行研究

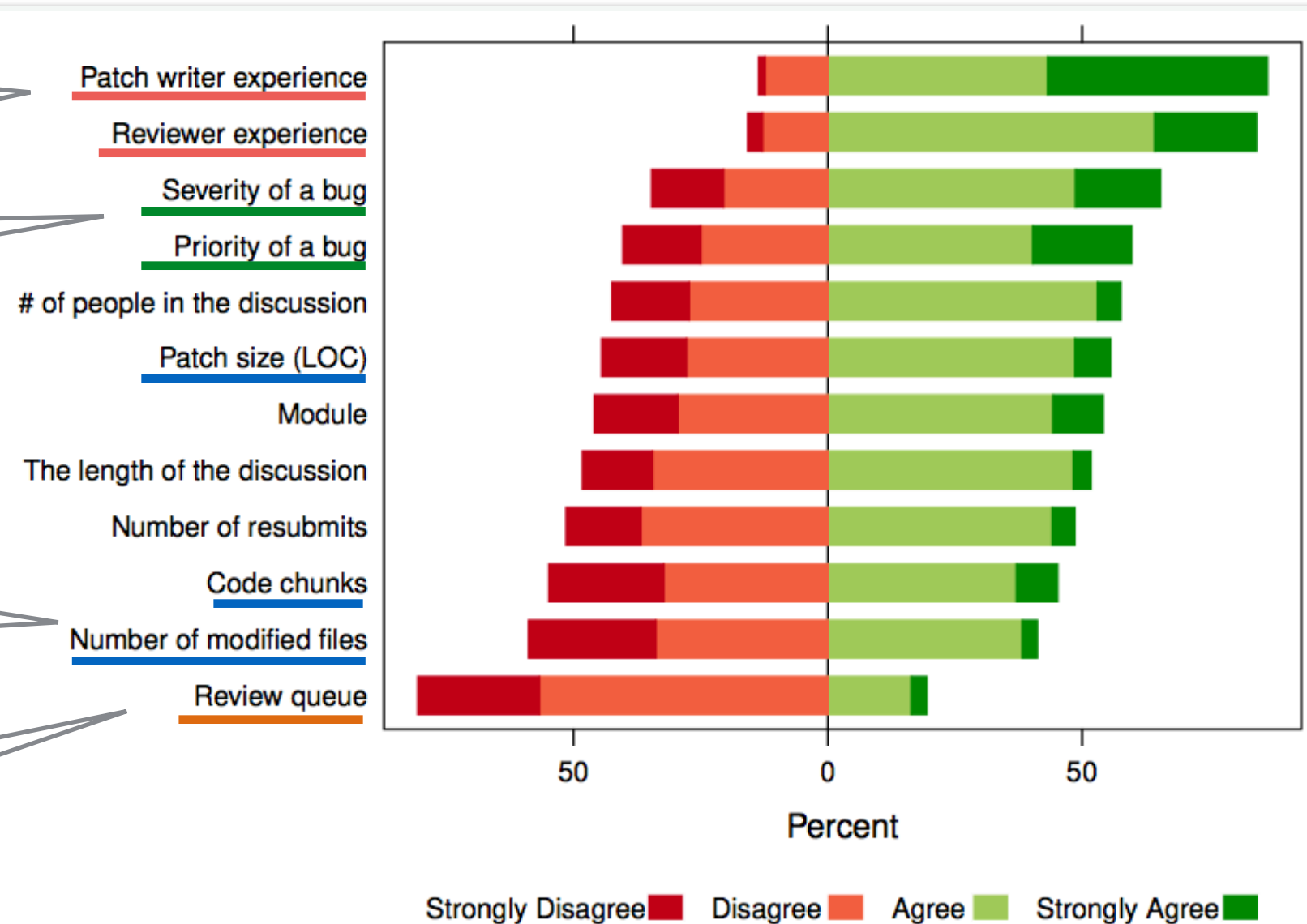


Figure 2: Factors influencing code review decision.

結果：RQ2, 開発者が考えるレビューの採否に影響する要因

Category	Q11	
	#Q	#R
Code quality	49%	57%
Testing	13%	28%
Time constraints	1%	4%
Change scope/rationale	11%	22%
Understanding code change/base	—	—
Human factors	—	—
Tools	—	—
Communication	—	—
Change complexity	—	—
Relationship/trust	5%	9%
Usefulness	—	—
Workload	—	—
Submitter related	2%	6%
Architecture/design	—	—
Reviewer related	9%	13%
Discussion	1%	4%
Conformance to project goals	6%	7%
Bug type	—	—
Selecting correct reviewer	—	—
Performance	—	—
Integration into code base	—	—
Security	—	—
Memory management	—	—
Familiarity with the author	—	—
Thorough feedback	—	—
Catching bugs	—	—
Organizational factors	—	—
Documentation	—	—
Context switch	—	—
Irrelevant	3%	7%
Total:	141	54
Average percent agreement	97.2%	

コードの質：◎

正確さと保守性をコードの質と関連づけている
(コーディング規約, 可読性, 設計)

testing：○

レビュワーはパッチに対してテストを行う
“テストが通ったら自信を持って承認できる”

社会的な関係：○

“新しい開発者は欠陥を含みやすいと思う”

Notes: #Q: the number of quotes, #R: the number of respondents, Q11: factors affecting decision, Q13: factors affecting time, Q14: patch quality, Q15: characteristics of code review quality, Q17: other factors affecting review quality, Q18: challenges.

結果：RQ3.1 いいパッチ & RQ3.2 いいレビュー

Category	Q14		Q15	
	#Q	#R	#Q	#R
Code quality	31%	65%	9%	22%
Testing	12%	36%	7%	15%
Time constraints	—	—	8%	20%
Change scope/rationale	26%	58%	—	—
Understanding code change/base	—	—	21%	30%
Human factors	—	—	17%	28%
Tools	—	—	—	—
Communication	—	—	—	—
Change complexity	10%	31%	—	—
Relationship/trust	—	—	—	—
Usefulness	1%	3%	1%	3%
Workload	—	—	—	—
Submitter related	—	—	—	—
Architecture/design	5%	15%	6%	10%
Reviewer related	—	—	—	—
Discussion	—	—	—	—
Conformance to project goals	—	—	—	—
Bug type	—	—	—	—
Selecting correct reviewer	—	—	—	—
Performance	2%	8%	—	—
Integration into code base	4%	15%	—	—
Security	1%	3%	—	—
Memory management	1%	2%	—	—
Familiarity with the author	1%	5%	—	—
Thorough feedback	—	—	23%	38%
Catching bugs	—	—	4%	8%
Organizational factors	—	—	—	—
Documentation	—	—	—	—
Context switch	—	—	—	—
Irrelevant	5%	15%	4%	10%
Total:	290	86	219	86
Average percent agreement	95.5%		94.2%	

RQ3.1 パッチの質

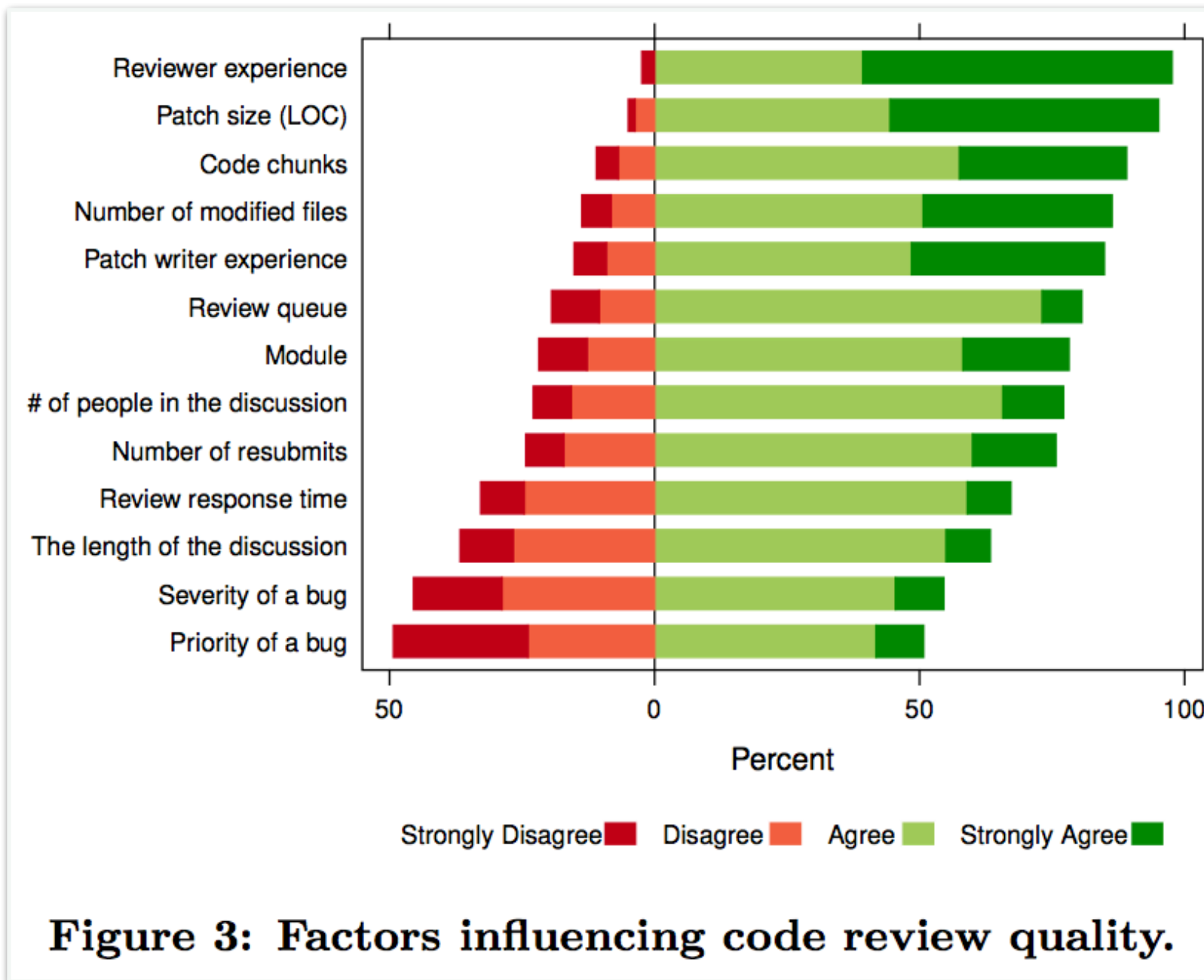
コードの質：可読性, 完結さ, 保守性
 変更割合：必要な箇所の変更
 複雑度：シンプルな変更

RQ3.2 良いレビューの特徴

わかりやすいフィードバック：フォー
 マットだけじゃない説明
 コードへの理解
 個人的要因：サポートしてくれる

Notes: #Q: the number of quotes, #R: the number of respondents, Q11: factors affecting decision, Q13: factors affecting time, Q14: patch quality, Q15: characteristics of code review quality, Q17: other factors affecting review quality, Q18: challenges.

結果：RQ3.3, 開発者が考えるレビューの質に影響する要因



RQ3.3

- ・ パッチサイズ：○
- ・ レビューワの個人的な要因：◎
 - RQ3.2
 - 応答性
- ・ 緊急性, 優先度：×

レビューの質は個人的な要因と強い関係がある

結果：RQ4, 開発者が直面するレビュー時の課題

Category	Q18	
	#Q	#R
Code quality	1%	2%
Testing	—	—
Time constraints	17%	25%
Change scope/rationale	10%	15%
Understanding code change/base	31%	38%
Human factors	11%	16%
Tools	9%	9%
Communication	1%	2%
Change complexity	10%	15%
Relationship/trust	—	—
Usefulness	—	—
Workload	4%	5%
Submitter related	—	—
Architecture/design	—	—
Reviewer related	—	—
Discussion	—	—
Conformance to project goals	—	—
Bug type	—	—
Selecting correct reviewer	—	—
Performance	—	—
Integration into code base	—	—
Security	—	—
Memory management	—	—
Familiarity with the author	—	—
Thorough feedback	—	—
Catching bugs	—	—
Organizational factors	—	—
Documentation	—	—
Context switch	6%	10%
Irrelevant	—	—
Total:	81	55
Average percent agreement	95.0%	

技術的な課題

コードの理解：何をしようとしているのか
 コードの複雑度：長いパッチはレビューが大変
 ツール：どこを変更したのかラインごとにわかる
 のはいいけど、何のパッチか要約が欲しい

個人的な課題

時間管理：どのパッチを優先的にレビューしたら
 いいのかわからない
 技術的な能力：新しい貢献者をサポートしたいけ
 ど、バグを含む可能性が上がるから、自分のスキル
 も磨かないといけない

Notes: #Q: the number of quotes, #R: the number of respondents, Q11: factors affecting decision, Q13: factors affecting time, Q14: patch quality, Q15: characteristics of code review quality, Q17: other factors affecting review quality, Q18: challenges.

まとめ

- ・ 大規模なプロジェクト（Mozilla）でコードレビューの質に対する**開発者の認識**について質的な調査をした
 - 時間と採否, レビューの質, 課題
- ・ 主な知見
 - **レビューの質**はフィードバックの内容や, レビューワーカーのコードへの精通, コードの質への理解と関係がある
 - **開発者**は時間の管理, 技術的な能力が課題だと感じている

所感

- ・ 開発者のレビュー時に重要視することがわかった
- ・ インタビュー内容が載っていない（多分公開してない）のが残念
 - どんな質問をしたのか
- ・ 既存研究をもとに, 同意する / しない を開発者に聞くのは面白いと思った

所感

- ・ Mozillaの人レビューツール使ってなくてショック
 - 見逃しバグの原因調査なので, Mozillaの人がレビューツール使おうが使わまいが本質は関係ないと考えているのですが...
- ・ 参考文献の既視感半端ない (英語から現実逃避してました)
 - Dabish, Apache v.s. Mozilla, Pull-based hoge Integrator's ...
 - ・ Rockstarのときの内容感
- ・ 参考文献の大事さを少し実感 (今更)
 - 位置づけがなんとなく見えてくる
 - 輪講で聞いたことあるタイトル?