

Process Aspects and Social Dynamics of Contemporary **Code Review**: Insights from Open Source Development and Industrial Practice at Microsoft

開発者のコードレビューへの認識をインタビュー調査

Amiangshu Bosu



Jeffrey C. Carver



Christian Bird



Jonathan Orbeck



Christopher Chockley



Industrial



OSS



背景

- ◎ コードレビューは幅広く使われている
- ◎ ツールベースの現代のコードレビューの効果を知りたい
- ◎ 商用（Microsoft）とOSSで差分があるのか知りたい

目的

- ◎ コードレビューに対する開発者の認識
- ◎ コードレビューで協力する理由
- ◎ コードレビューにおける技術的でない側面
- ◎ OSSとMicrosoftのコードレビューの比較

◎ コードレビューの重要性

- ▶ RQ1 コードレビューが重要だと考える理由

◎ コードレビュープロセス

- ▶ RQ2 コードレビューに使う時間
- ▶ RQ3 コードレビュー要求を受け入れる理由
- ▶ RQ4 品質の低いコードの特徴
- ▶ RQ5 品質の低いコード作成者への支援方法

◎ レビューワート士の印象

- ▶ RQ6, RQ7 高品質 / 低品質 なコード作成者への印象

対象

- ◎ 十分なコードレビュー経験を持つ開発者
 - ▶ 30件以上のコードレビューを行なった開発者

- ◎ プロジェクト
 - ▶ 34種類のOSS
 - ▶ Microsoft

- ◎ 調査対象開発者
 - ▶ OSS
 - 287人 (16%, 1,815人中)
 - ▶ Microsoft
 - 416人 (21%, 2,000人中)

アプローチ

- ◎ インタビュー
 - ▶ 回答者の内訳

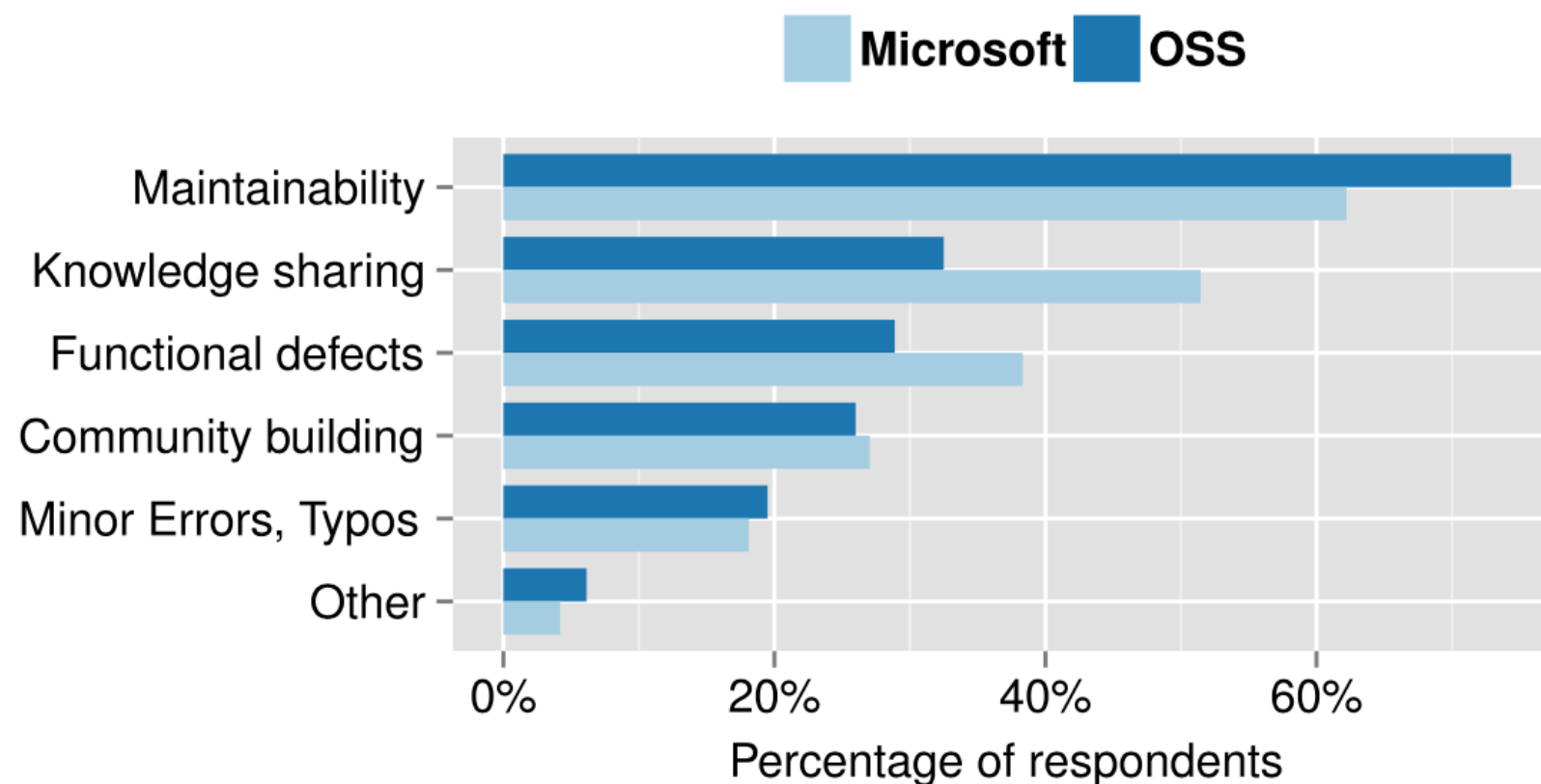
TABLE 4
Respondents' Primary Projects

Open Source Projects			Microsoft Projects		
Qt Project (36)	OpenStack (32)	CyanogenMod (28)	Bing (54)	Windows (48)	Office (43)
TYPO3 (20)	Android (19)	MediaWiki (17)	Azure (24)	Visual Studio (22)	XBox (13)
oVirt (17)	Linux kernel (16)	Chromium OS (14)	Ad Center (11)	Exchange (10)	Dynamic AX (12)
Eclipse (9)	Gromacs (9)	ITK (9)	Windows Phone (10)	IE (6)	SQL Server (6)
LibreOffice (8)	OpenAFS (6)	Scilab (5)	Dynamic CRM (4)	Sharepoint (4)	Skype (3)
VTK (5)	Gerrit (4)	AOKP (2)	Halo (3)	TFS (3)	HPC (3)
Couchbase (2)	Debian (2)	Others (24)	Autopilot (3)	Lync (3)	Others (131)

- ◎ インタビュー内容の妥当性を検証
 - ▶ 尺度項目（質問内容, 内容の有効性, 尺度項目の有効性）
 - ▶ 自由回答はグランデッドセオリー

結果RQ1：コードレビューが重要だと考える理由

- ◎ コードレビューが重要だと考えている開発者
 - ▶ OSS：98.6%
 - ▶ Microsoft：100%
- ◎ 重要だと考える理由



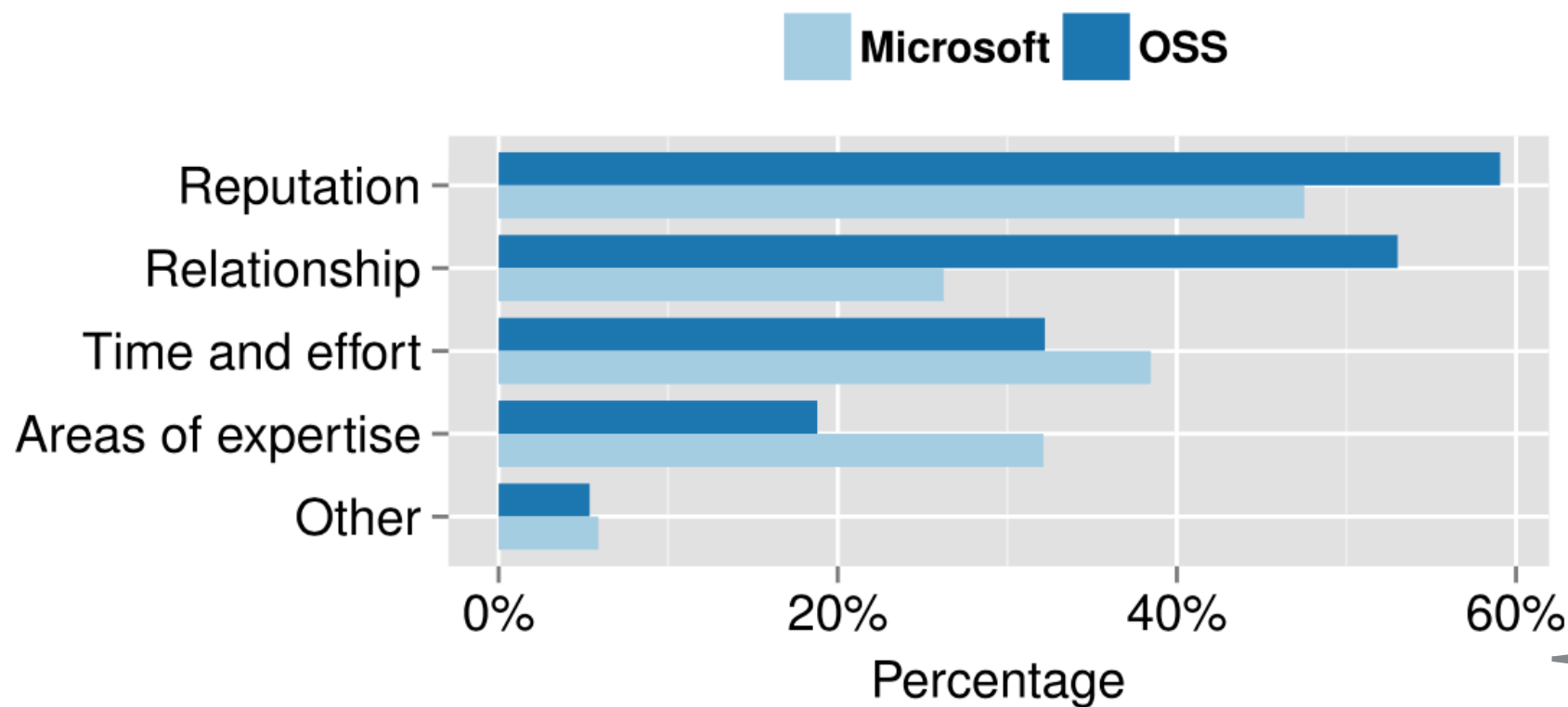
保守性の向上
知識の共有
欠陥の排除
共同作業を促す
タイプミスの特特定

結果RQ2：コードレビューに使う時間

- ◎ 開発者がコードレビューに使う時間
 - ▶ OSS：5時間/週
 - ▶ Microsoft：4時間/週
- ◎ 作業時間週40時間とすると, 10-15%の時間を使っている

結果RQ3：コードレビュー要求を受け入れる理由

- ◎ 著者のidentityが重要
 - ▶ OSS：半数以上
 - ▶ Microsoft：2/3
- ◎ 具体的な理由



OSS開発者

評判と関係性を重視

Microsoft開発者

技術的要因を重視
(時間, 労力, 専門知識)

結果RQ4：品質の低いコードの特徴

- 先行研究から低品質なコードの特徴10個
 - ▶ 開発者にランク付けしてもらう

可読性が大切

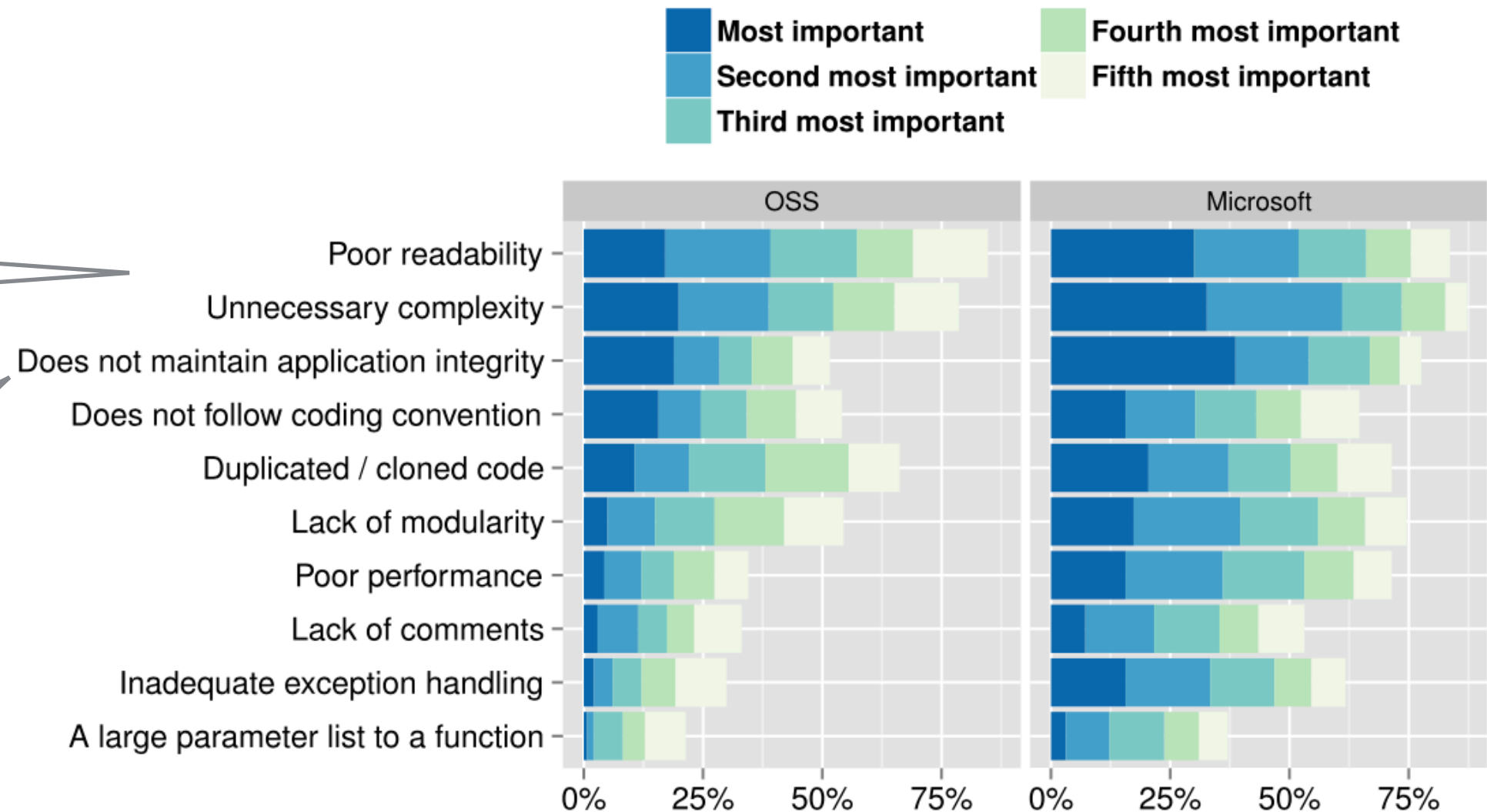
コメントはそれほど重要ではない

アプリケーションとの整合性

長期的なプロジェクトには重要

コーディング規約

OSS（4番目）と Microsoft（8番目）で認識に差があった

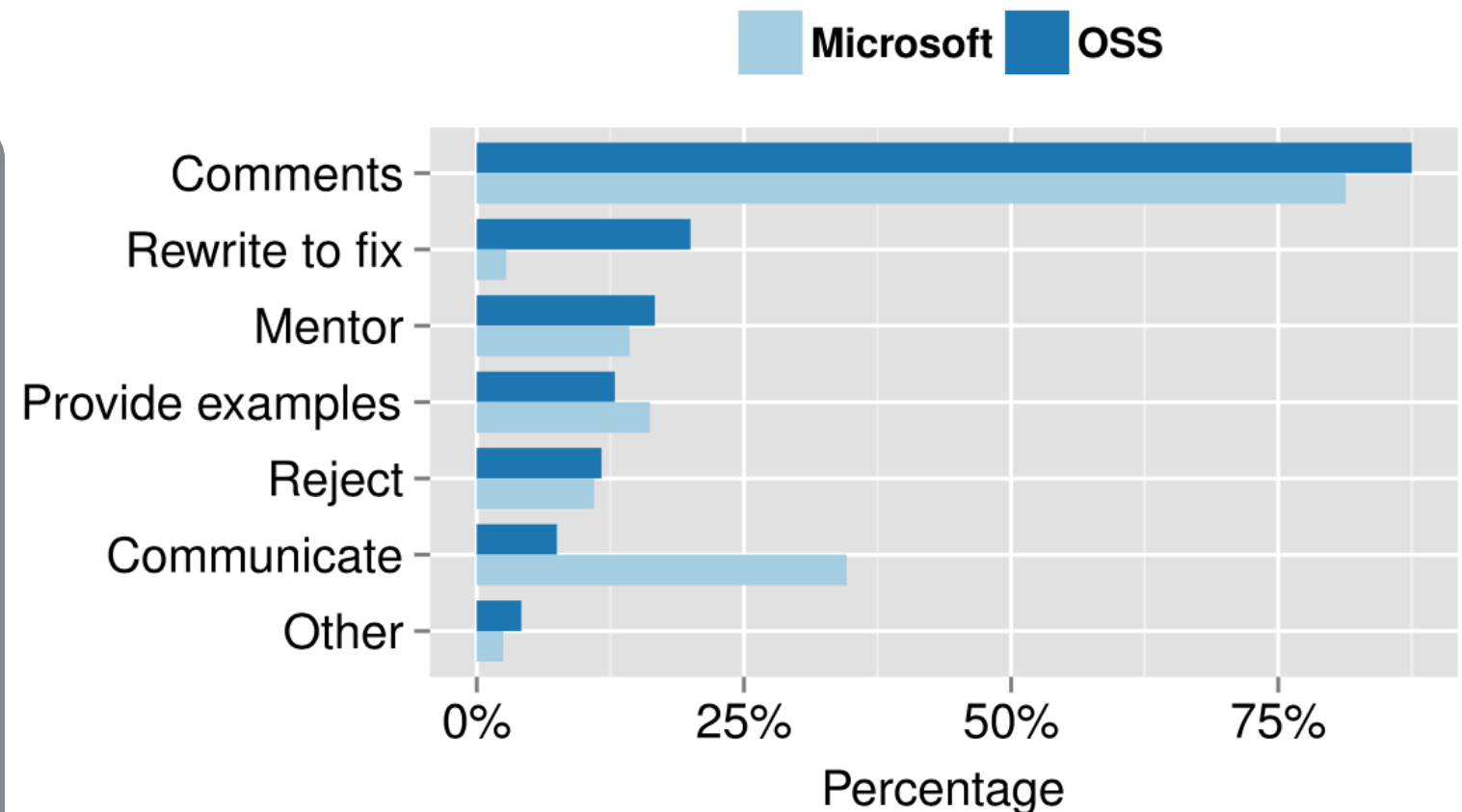


結果RQ5：品質の低いコード作成者への支援方法

- 回答者の80%以上がコードの改善を支援している

コメントを提供
コードを修正
メンターを提供
例を提供

十分な品質になるまで拒否
著者とコミュニケーション



結果RQ6：高品質なコード作成者への印象

- ◎ コードの品質は作成者の印象に影響を与える（85%以上）
 - ▶ 印象が上がる（有能, コア開発者の兆候）
 - OSS：60%
 - Microsoft：36%
 - ▶ 関係を構築したいと感じる
 - OSS：37%
 - Microsoft：47%
 - ▶ 将来, コラボレーションをしたいと感じる
 - OSS：36%
 - Microsoft：48%

考察とまとめ

◎ コードレビューの認識

▶ OSS開発者

- お互いの印象形成（関係の構築）
- コード作成者との関係とコード作成者の評判を重要視

▶ Microsoft

- 知識の共有のために重要（チームメイトの指導）

◎ コードレビューの利点

▶ 欠陥削減だけではない（欠陥削減目的は全体の1/4）

- 知識の普及, 関係構築, 保守性の向上

所感

- ◎ インタビュー項目自体を評価する方法があることに驚いた
- ◎ OSSは人間性（評判）を重要視しているけど, Microsoftはそうでもない