

On the Naming of Methods: A Survey of Professional Developers

Reem S. Alsuhaibani, Christian D. Newman, Michael J. Decker,
Michael L. Collard, Jonathan I. Maletic

ICSE 2021 Technical Track “On the Naming of Methods: A Survey of Professional Developers”

A solid orange horizontal bar at the bottom of the slide.

背景：ソースコードの識別子

- 識別子の命名はソフトウェア工学において非常に重要.
 - ソフトウェアの可読性と理解力に影響がある. [2-6]
- 命名に気を遣うことで起きるメリット
 - ソフトウェアのメンテナンスコストを削減できる. [7, 8]
 - 開発者に高いレベルのタスクの意図を伝えることができる. [9]
- 命名の仕方には多くの暗黙の了解と文書化された基準がある.
- これらが開発者に受け入れられているかは広く理解されていない.

[9] J. I. Maletic and A. Marcus, “Supporting program comprehension using semantic and structural information,” in Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001, 2001, pp. 103–112.

本研究の目的

- ソフトウェア工学の文献に記載されている名前のつけ方の推奨が、実際の実務を反映しているか、開発者の意見と一致しているかの判断.
- 開発者に広く受け入れられているメソッドの命名基準を伝えること.

先行研究

- 品詞列のレベルで分析[12]
 - 関数名が最も多くのユニークな名前のバリエーションを持つことがわかって
いる.
 - メソッド名が他のタイプの識別子より複雑である.
- これから考えられること
 - メソッド名に焦点を当てた調査の優先が研究コミュニティに最も貢献する.
 - ほかのタイプの識別子に関する将来の研究の動機づけになる.

[12] C. D. Newman et al., “On the generation, structure, and semantics of grammar patterns in source code identifiers,” J. Syst. Softw., vol. 170, p. 110740, Dec. 2020.

RQ(リサーチクエスチョン)

- RQ1

- ソフトウェア専門家は、調査で伝えられたメソッドの命名基準をどの程度支持しているか。(開発者は基準の各部に同意しているか、基準は完全か)

- RQ2

- プログラミングの経験年数は、アンケートの回答に影響を与えるか。

- RQ3

- プログラミング言語によってメソッドの命名の仕方に影響が出るのか。

- RQ4

- 本調査におけるメソッド命名基準の各項目について、専門家はどう認識しているか。命名基準の各項目に関する好みと障壁は何か。

メソッド命名基準

- 関連するソフトウェア工学の文献と既存のコーディング基準を調査して作成.
- 上記知見から10の基準を特定.
 1. 命名規則
 2. 文法構造
 3. 動詞句
 4. 辞書の用語
 5. 全文
 6. イディオム・スラング
 7. 略語
 8. 頭字語
 9. プリフィクス・サフィクス
 10. 長さ

メソッド命名基準

- 命名規則
 - 開発者は主に2つの命名スタイルをアプリケーション内で一貫して使う.
 - キャメルケース (ex : getFullName())
 - スネークケース (ex : call_with_default())
 - 詳細な研究[50]では違いは特にないとされている.
 - Binkleyらの研究[51]では, キャメルケースの方が精度が高かった.
 - プログラマーではない人は, キャメルケースは視覚的に処理しにくい.

[50] D. Binkley, M. Davis, D. Lawrie, J. I. Maletic, C. Morrell, and B. Sharif, “The Impact of Identifier Style on Effort and Comprehension,” *Empir. Softw. Eng.*, vol. 18, no. 2, pp. 219–276, Apr. 2013.

[51] D. Binkley, M. Davis, D. Lawrie, and C. Morrell, “To camelcase or under_score,” in *2009 IEEE 17th International Conference on Program Comprehension*, 2009, pp. 158–167.

メソッド命名基準

- 文法構造
 - 複数の単語を含むメソッド名は、文法的に正しい文構造をもつ必要がある。

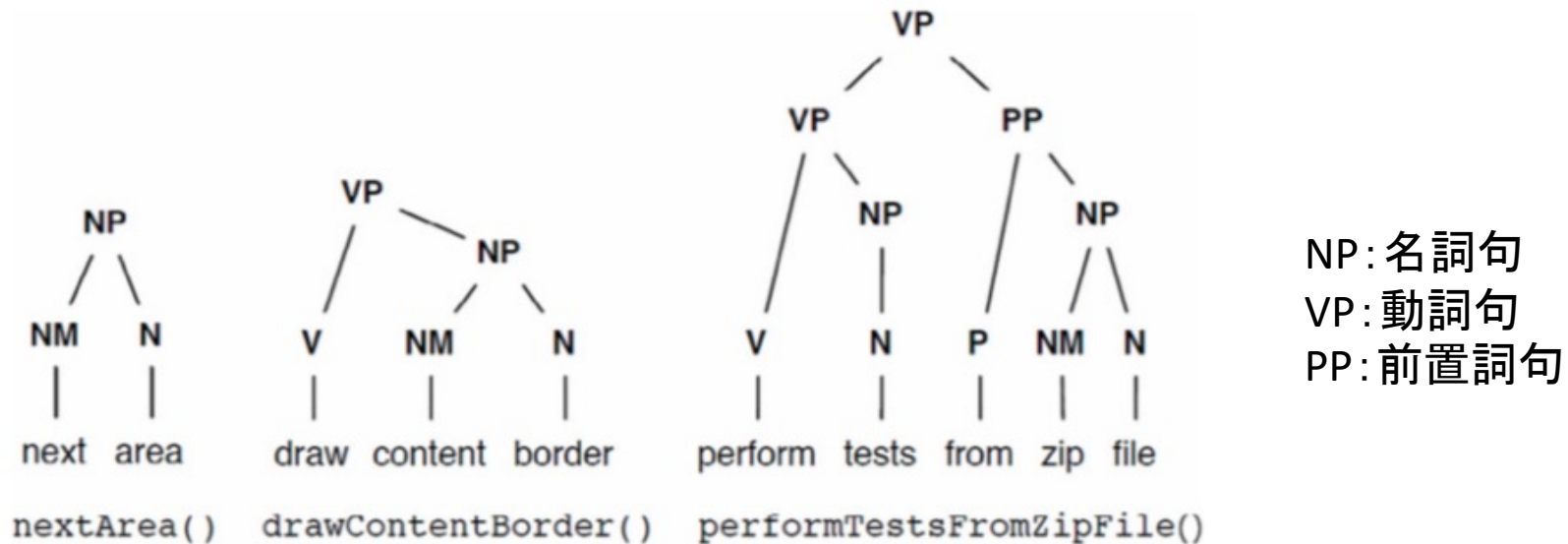


図1: 3つのメソッド名識別子の名詞句、動詞句および前置詞句を伴う動詞句の例.

メソッド命名基準

- 動詞句

- メソッド名には、常にメソッドの動作を示す動詞(複数可)または動詞句を含める必要がある。

- 辞書の用語

- メソッド名に使用される単語が実際の辞書用語である。

- 全文

- メソッドのタスクを明確に示すため、メソッド名には1文字ではなく完全な単語を使用する必要がある。

メソッド命名基準

- イディオム・スラング
 - メソッド名には、個人的な表現や慣用句、未知のスラングを含んではならない。
- 略語・頭字語
 - 既知または標準的な略語または頭字語のみ使用できる。複数の解釈が生まれるものは不適切である。

メソッド命名基準

- プリフィクス・サフィクス
 - システムからの用語である接頭辞/接尾辞を含んではならない。ただしC言語を除く。
- 長さ
 - メソッドの長さについて。

調査設計・方法

- Kitchenham[82]のソフトウェア工学における個人意見調査のガイドラインに従い、以下の2つを調査の目的とした。
 1. 何が良いメソッド名となるのかについての全体的な合意.
 2. プログラミング年数とプログラミング言語の背景に基づく、各メソッド命名基準に対する開発者の意識.

[82] B. A. Kitchenham and S. L. Pfleeger, “Personal Opinion Surveys,” in Guide to Advanced Empirical Software Engineering, F. Shull, J. Singer, and D. I. K. Sjøberg, Eds. London: Springer London, 2008, pp. 63–92.

調査設計・方法

- 調査のためにQualtrics(qualtrics.com)を用いたアンケートを実施.
 - 各方式の命名基準に関する10の調査質問と参加者の情報を収集する人口統計学的質問の主に2つの質問を用意.
- 専門家が基準についてどう考えているかを調査するため、各基準を取り上げ、選択式と自由解答形式の調査質問も実施.
 - 基準の適応可能性と参加者の同意又は非同意について専門家からフィードバックを得るのが目的.
- 人口統計学的質問は、異なる背景を持つ専門家が基準をどの様に認識しており、より多くの洞察を見出すのに役立つ.

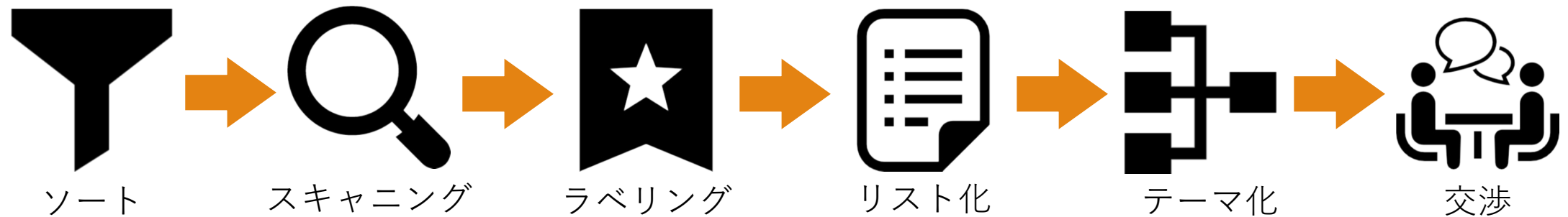
調査方法・設計

- 調査参加者について
 - Google, IBM, Intel, Microsoftなどの大手ソフトウェア企業に勤める専門家に、個別でメールで招待。 [36]
 - ソフトウェア工学の研究者に個別に参加を呼びかけ、さらに他の専門家にも招待状を転送する様依頼。
 - Stack Overflowコミュニティにおいて共有。

[36] T. Punter, M. Ciolkowski, B. Freimut, and I. John, “Conducting online surveys in software engineering,” in 2003 International Symposium on Empirical Software Engineering, 2003.
ISESE 2003. Proceedings., Rome, Italy, 2003, pp. 80–88.

データ収集と分析

- 調査にはQualtricsを用いている.
- 自由形式の質問はMAXQDA(maxqda.com)を用い、テーマ別コーディングを行った.



図：各規格に対する参加者のコメントを分析する手順

データ収集と分析

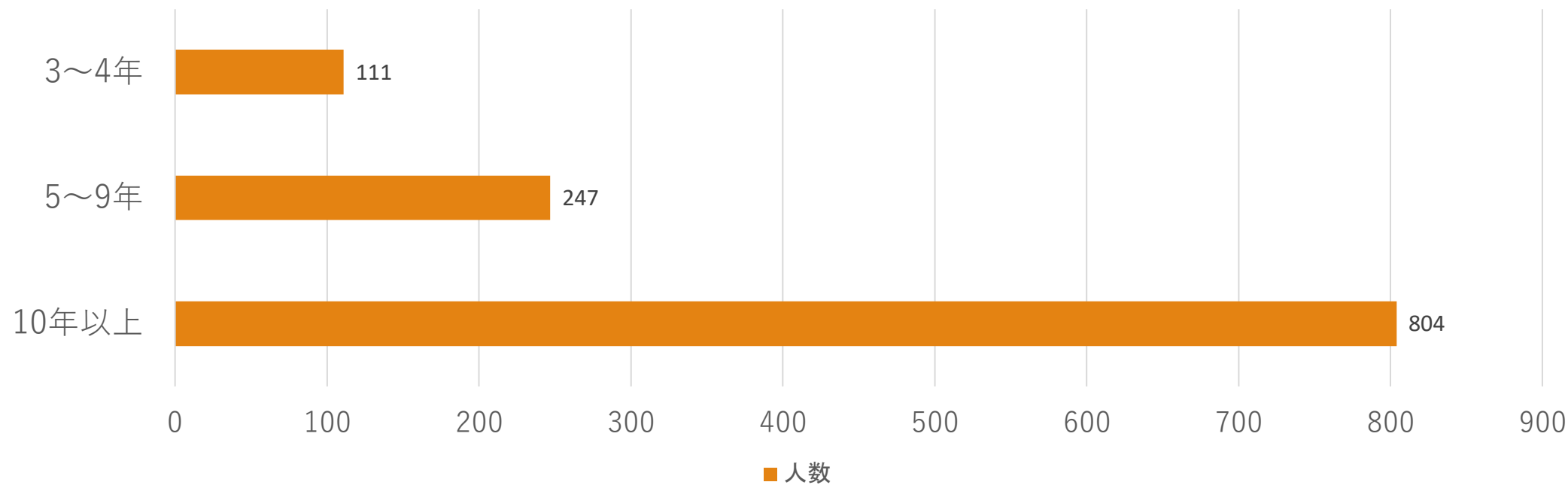
- まず経験年数やプログラミング言語の知識に基づいてコメントを整理.
- コメントの全体的な傾向を把握するために, コメントを読み込む.
- コメントをある程度理解した上で, 一般的なトピックのラベル(カメラケース, 可読性, テストなど)を付与.
- 以上が完了するとすべてのラベルのリストが作成される.
- このリストは, テーマのセットを作成するために使用される. 関連するコメントは全てラベルに基づきグループ化される.
- またほかの研究者ともテーマについて交渉, 必要に応じテーマを微調整.

結果

- 1604通の回答を取得.
- 近年発表された他の調査ではでは820件^[85], 126件^[86]となっており, 今回の回答は非常に優れた回答であると言える.
- 分析の際は不完全な回答を除去し, 経験年数3年以上の実務家の解答のみを考慮.

結果

● 経験年数別の参加者総数



結果

●1164人の参加者が使用したプログラミング言語

言語	回答数	その言語を知っている参加者の割合
C++	472	40.6%
C	484	41.7%
C#	589	50.7%
Python	635	54.7%
Java	697	50.0%
JavaScript	838	72.1%
他言語	479	41.2%
総回答数	4194	
平均言語数	4	

結果

- 経験年数10年以上が70%と大半を占めており、経験豊富な開発者を調査する目的は達成された。
- 参加者の40～50%がC++、C、C#、Python、Javaを知っており、**言語背景には素晴らしい多様性がある**といえる。
- 約40%の参加者がリストにない他の言語の経験もあると回答、その中でもPHP、Ruby、Goはそれぞれ100人程度の参加者が知っていると回答。
- さらにその他の言語(35の言語)を知っている参加者の数は50人以上はいない。(多くは15人)
- 平均すると一人当たり4つのプログラミング言語を知っていることになる。

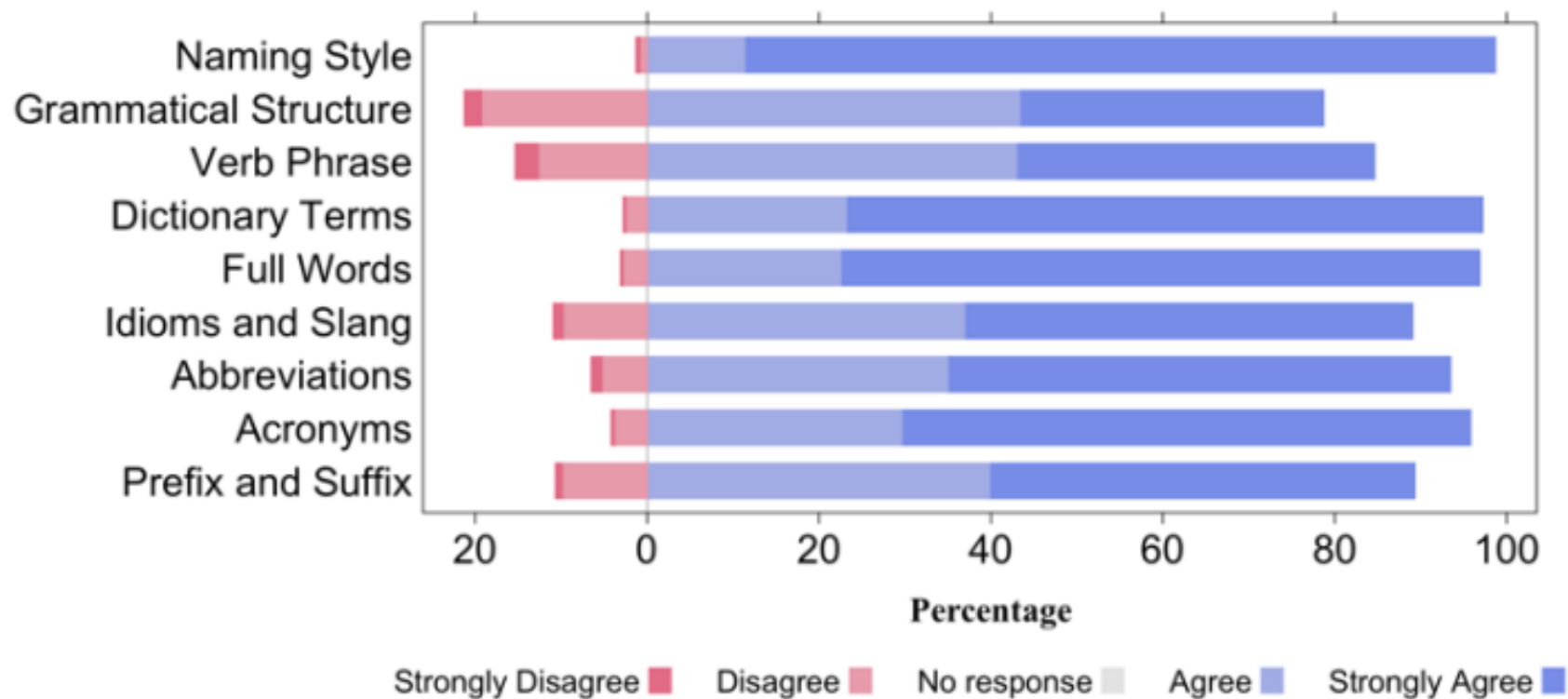
結果

●各規準に関する10の調査項目に対する回答

Question(質問)	Response Type(回答タイプ)			
	強く同意する	同意する	同意しない	強く反対する
命名規則	87.20%	11.51%	0.87%	0.43%
文法構造	35.21%	43.51%	19.29%	1.99%
動詞句	41.50%	43.14%	12.68%	2.67%
辞書の用語	73.89%	23.33%	2.52%	0.26%
全文	74.20%	22.69%	2.85%	0.26%
略語	58.33%	35.16%	5.30%	1.22%
頭字語	66.03%	29.79%	3.83%	0.35%
イディオム・スラング	51.99%	37.09%	9.79%	1.13%
プリフィクス・サフィクス	49.34%	39.98%	9.89%	0.79%

結果

- 各規準に関する10の調査項目に対する回答

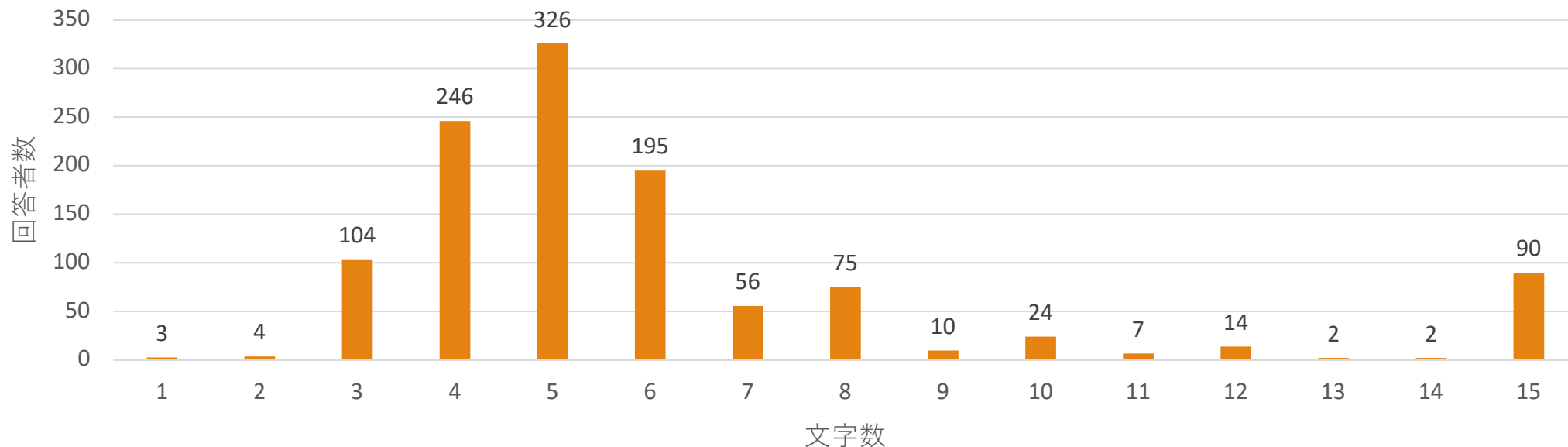


結果

- 全体として、すべての基準について、「同意する」以上の回答が大半を占めており、多くが90%以上である.
- 「動詞句」と「文法構造」の2つの基準のみ、「同意しない」以下の回答が顕著である.
- 「長さ」はリッカート尺度によらないためここでは割愛.

結果

- 「長さ」について
 - 中央値，最頻値共に5語で， $5\text{語} \pm 2\text{語}$ 以下で見ると81%もの回答がある.
 - 比較的短いメソッド名であるべきというのが非常に強い共通認識となっている.



RQの回答

- RQ1

- ソフトウェア専門家は、調査で伝えられたメソッドの命名基準をどの程度支持しているか。（開発者は基準の各部に同意しているか、基準は完全か）

- RQ2

- プログラミングの経験年数は、アンケートの回答に影響を与えるか。

- RQ3

- プログラミング言語によってメソッドの命名の仕方に影響が出るのか。

- RQ4

- 本調査におけるメソッド命名基準の各項目について、専門家はどう認識しているか。命名基準の各項目に関する好みと障壁は何か。

RQの回答

- RQ1

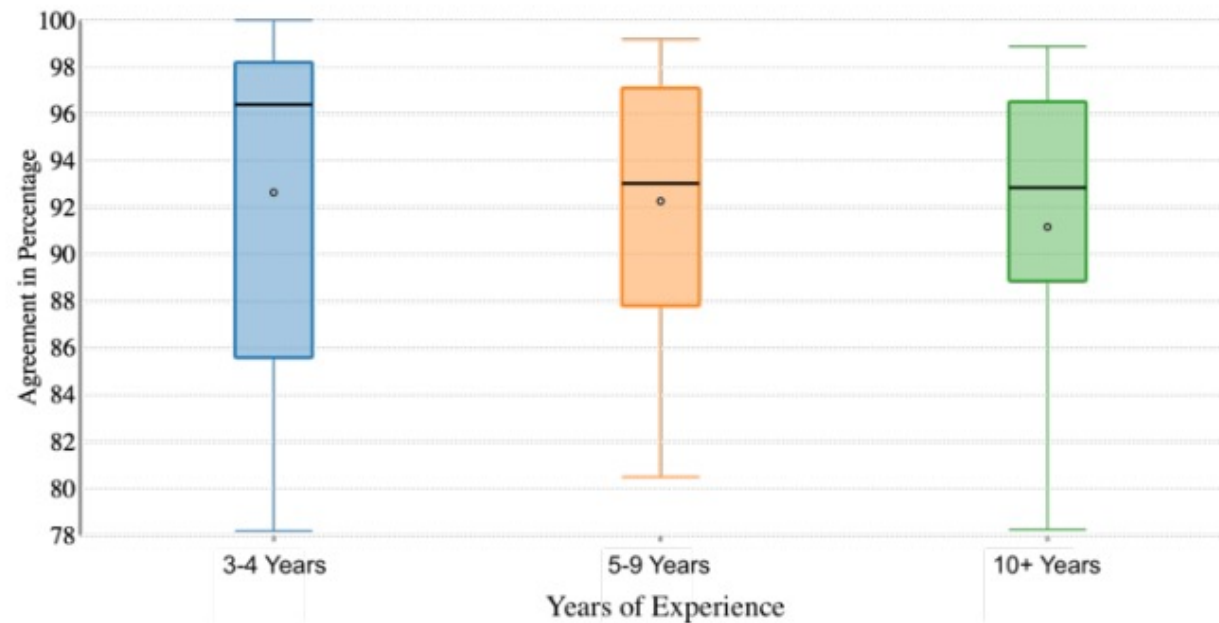
- ソフトウェア専門家は、調査で伝えられたメソッドの命名基準をどの程度支持しているか。（開発者は基準の各部に同意しているか、基準は完全か）
- 提示された手法の命名基準には幅広い支持がある。
- 開発者のプログラミング言語の背景知識の多様性を考慮すると、この結果はほとんどのプログラミング言語に一般化されると思われる。

RQの回答

- RQ2
 - プログラミングの経験年数は、アンケートの回答に影響を与えるか.
- 与えない.
- 個々の質問についてカイ二乗検定を行ったところ、略語のみ有意差があった.
 - 経験の浅い開発者より略語の使用に慣れているため差があるとみられる.
 - 経験を積むことでより多くの略語に精通し、その拡張より略語を好むという仮説を立てた.

RQの回答

- すべてのグループ参加者の平均値と中央値が一致している。

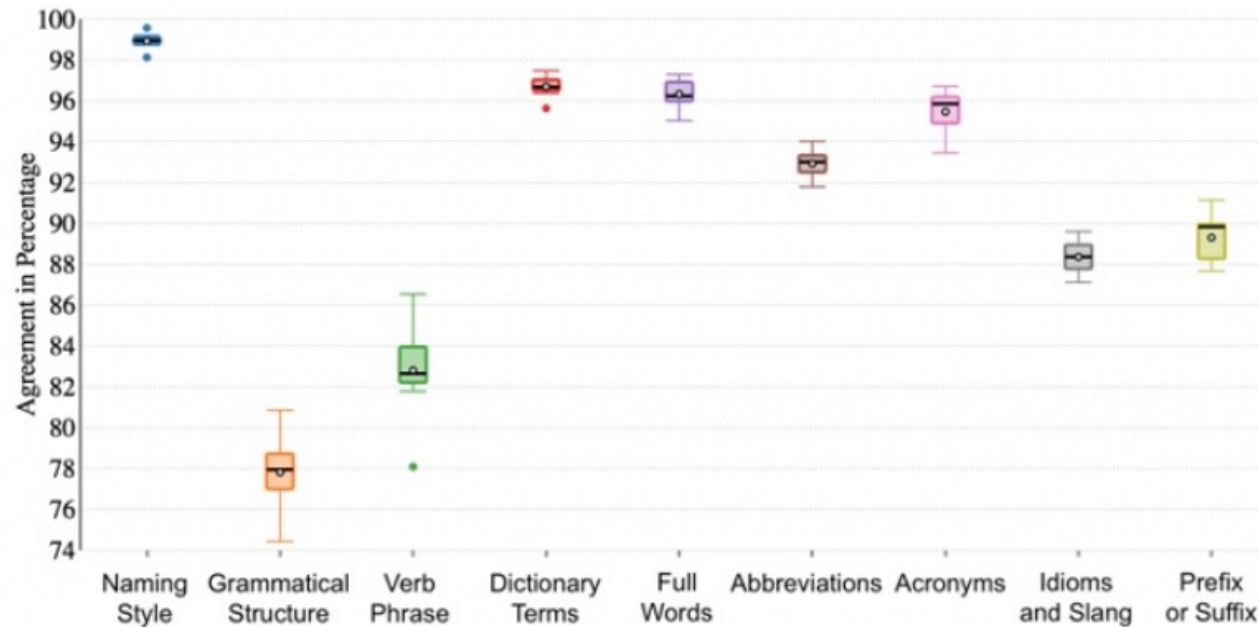


図：経験年数別の全標準の一致率

RQの回答

● RQ3

- プログラミング言語によってメソッドの命名の仕方に影響が出るのか.



図：各質問の一致率

RQの回答

- RQ3
 - プログラミング言語によってメソッドの命名の仕方に影響が出るのか.
- 各質問の標準偏差から外れたものはほとんどない.
- 言語経験の有無にかかわらず，誰もが非常に似たような回答をしている.
- 外れ値について
 - 命名規則：JavaとC#の経験を持つ開発者(少数)
 - 辞書の用語：Cの経験を持つ開発者(少数)
 - 動詞句：多言語のカテゴリから.

RQの回答

- RQ4

- 本調査におけるメソッド命名基準の各項目について，専門家はどう認識しているか．命名基準の各項目に関する好みと障壁は何か．
- 概ね受け入れている．
- 命名規則がどの程度厳格に守られているかを尋ねる質問では，6割が厳格な命名基準を設けていないことがわかった．
- いくつかの標準規格には多少の柔軟性が必要であること以外は，好みや障壁はなかった．

有効性

- 質問の構成要素の妥当性[90]に対処するため，専門家とディスカッションを実施，質問内容が明確であることを確認.
- 調査における規格は，ソフトウェア工学の文献や公表されているコーディング規格から得ている.
- メソッドの命名品質を調べる質問のみを考慮することで，調査の内部一貫性に対処.
- 参加者の出身国は主に欧州と北米の計72ヶ国であったため，幅広い層に対して一般化できる結果といえる.

[90] M. S. Litwin and A. Fink, The survey kit. 7: How to measure survey reliability and validity, 4. print. Thousand Oaks, Calif.: Sage, 1997.

結論

- 開発者は与えられた規格に非常に同意している.
 - メソッドの命名基準を明確にすることを支持し、それがコードの理解にいい影響を与えていると感じている.
- メソッドの命名基準が有効であり、少なくとも開発者には広く受け入れられている.
- この規格は完全である.

展望

- メソッド名の品質を高い精度で評価する自動ツールを構築するためにこれらの規格を使用できる.
- 自然言語処理ツールと辞書を使用すれば, このようなツールに規格のすべての側面をかなりの程度まで実装できる.
- このようなツールは開発者やコードレビューの際に非常に有用で, メソッド名の推薦システムにもつながる.

調査データ

- メソッドの命名基準に関するプロの開発者の意見を調査したデータをGitHubの[91]で再現研究のため公開されている.
- データセットはMendeley Data Archive Repositoryにて公開されている.

[91] R. S. AlSuhaibani, C. D. Newman, M. J. Decker, M. L. Collard, and J. I. Maletic, “Method Naming Standards Artifact,” GitHub Repos., 2021. <https://github.com/KSU-SDML/Method-Naming-StandardsArtifact>

[92] R. S. AlSuhaibani, C. D. Newman, M. J. Decker, M. L. Collard, and J. I. Maletic, “Method-Naming-Standards-Survey-Dataset,” Mendeley Data, vol. 1, 2021. <http://dx.doi.org/10.17632/5d7vx88sph.1>

所感

- 文書で書いてあることとデータの内容にずれがあった.
- 内部妥当性についての理解が難しく，読み解けなかった.

引用

- [2] A. A. Takang, P. A. Grubb, and R. D. Macredie, “The effects of comments and identifier names on program comprehensibility: an experimental investigation,” *J Prog Lang*, vol. 4, no. 3, pp. 143–167, 1996.
- [3] T. A. Corbi, “Program understanding: Challenge for the 1990s,” *IBM Syst. J.*, vol. 28, no. 2, pp. 294–306, 1989.
- [4] D. Lawrie, C. Morrell, H. Feild, and D. Binkley, “What’s in a Name? A Study of Identifiers,” in *14th IEEE International Conference on Program Comprehension (ICPC’06)*, 2006, pp. 3–12.
- [5] D. Lawrie, C. Morrell, H. Feild, and D. Binkley, “Effective identifier names for comprehension and memory,” *Innov. Syst. Softw. Eng.*, vol.3, no. 4, pp. 303–318, 2007.
- [6] S. Fakhoury, Y. Ma, V. Arnaoudova, and O. Adesope, “The effect of poor source code lexicon and readability on developers’ cognitive load,” in *Proceedings of the 26th Conference on Program Comprehension - ICPC ’18*, Gothenburg, Sweden, 2018, pp. 286–296.
- [7] B. W. Boehm, *Software Engineering Economics*, vol. 197. Prentice Hall PTR, 1981.
- [8] L. Erlikh, “Leveraging Legacy System Dollars for E-Business,” *IT Prof.*, vol. 2, no. 3, pp. 17–23, May 2000.
- [9] J. I. Maletic and A. Marcus, “Supporting program comprehension using semantic and structural information,” in *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001*, 2001, pp. 103–112.

引用

- [85] C. Brindescu, M. Codoban, S. Shmarkatiuk, and D. Dig, “How do centralized and distributed version control systems impact software changes?,” in Proceedings of the 36th International Conference on Software Engineering, Hyderabad India, 2014, pp. 322–333.
- [86] S. McKee, N. Nelson, A. Sarma, and D. Dig, “Software Practitioner Perspectives on Merge Conflicts and Resolutions,” in 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME), Shanghai, 2017, pp. 467–478.