Balancing Personality and Technical Expertise in Microservice Teams: An Algorithmic Approach

Dario Amoroso d'Aragona[1 [0000−0002−1363−2184]], Fabio Calefato[2 [0000−0003−2654−1588]], and Davide Taibi[3 [55920884000]]

[1] Tampere University, Finland
[2] University of Bari, Bari, Italy
[3] University of Southern Denmark, Vejle, Denmark
dario.amorosodaragona@tuni.fi, fabio.calefato@uniba.it, taibi@imada.sdu.dk

Abstract. Microservice architectures are commonly organized around small service-owning teams, making team composition a relevant architectural and human-factor concern. We propose an algorithmic approach for composing microservice teams by combining developers' Big Five personality traits, inferred from GitHub communication, with programming-language expertise extracted from contribution histories. The approach evaluates candidate assignments through a personality composition score grounded in organizational-psychology evidence and a technical-alignment score based on language coverage. A study on Spinnaker, HMDA-Platform, and Taskcluster shows that the method identifies higher-scoring candidate configurations and, more importantly, makes explicit how team size, contributor overlap, and technological diversity condition the feasible improvement space.

Keywords: Microservices · Team Composition · Big Five · Five Factor Model

1 Introduction

Microservices decompose systems into small, independently deployable services that are often owned by autonomous teams responsible for the full service lifecycle [7]. Because such teams are usually small, individual differences among developers may have stronger effects on coordination, communication, and conflict than in larger teams. Organizational psychology provides relevant evidence: high team-level *Conscientiousness* and *Agreeableness* are consistently associated with better team outcomes, while high *Neuroticism* is associated with poorer collaboration [1,13,3]; lower dispersion in key traits can also support coordination, although mean trait levels are usually more predictive than variance [13,15].

These findings have rarely been operationalized in microservice-oriented OSS settings, where membership emerges from contribution patterns rather than deliberate staffing. Li et al. [10] proposed personality-based collaboration optimization for microservice projects, but empirical evidence on real projects remained limited. We address this gap by proposing and evaluating a team-composition

approach that combines personality and programming-language expertise. We ask: **RQ1** whether algorithmic composition improves personality composition scores compared with existing configurations; **RQ2** how structural characteristics such as team size and developer distribution affect optimization; and **RQ3** how much technical alignment contributes to the combined score.

2 Background

This section summarizes the theoretical basis used to define the scoring function. The Big Five model describes personality through *Openness*, *Conscientiousness*, *Extraversion*, *Agreeableness*, and *Neuroticism* [8,6]. In software engineering, prior work has shown that Big Five traits can be estimated from textual traces such as mailing-list messages, issue comments, and pull-request discussions [16,2,5,4]. In team-performance research, meta-analyses identify *Conscientiousness* as the most robust predictor of team effectiveness and *Agreeableness* as the second most relevant trait for cooperation and cohesion [1,13,3,9]. Minimum *Agreeableness* is also relevant because one uncooperative member can disrupt group processes [1,15]. Effects for *Extraversion*, *Openness*, and *Neuroticism* are more context-dependent: *Extraversion* may support communication [12,3], *Openness* may support complex problem solving [13], and low *Neuroticism* may support emotional stability [1,14]. We operationalize these findings into an evidence-based composition quality score that allows alternative service-team assignments to be compared consistently.

3 Empirical Study Design

We evaluate the approach on three public GitHub microservice projects with different structures: *HMDA-Platform* (39 developers, 15 microservices, 2–4 developers per service, mean = 2.6), *Taskcluster* (27 developers, 12 microservices, 1–3 developers per service, mean = 2.25), and *Spinnaker* (273 developers, 12 microservices, 3–44 developers per service, mean = 22.75). This ordering is also used in the results to improve readability.

3.1 Data Collection and Personality Inference

For each microservice, we identified core developers from commit histories as the smallest contributor subset accounting for at least 80% of commits. If this produced too small a candidate pool, we added developers until reaching $3 \times k$ candidates, where k is the number of microservices. For each selected developer, we collected GitHub issues, issue comments, pull-request descriptions, and pull-request comments, and aggregated them into a developer-level textual corpus. Personality was inferred using seven pre-trained Big Five/OCEAN text models applied to three text variants: original text, regex-cleaned text with code/log artifacts removed, and grammar-corrected cleaned text. Rather than relying on

a single classifier, the inference step explicitly tests cross-model stability: for each model, we compute the mean pairwise prediction difference against the others, retain the three most stable models, and use majority voting to assign the final OCEAN label for each trait. This makes the subsequent optimization depend on consensus personality labels rather than on one model's isolated prediction.

Technical expertise was estimated from contribution histories. For each developer, language skill $skill(d, l) \in [0, 1]$ was computed from the proportion of contributions in language l. For each microservice, language requirements were extracted from repository structure and file extensions, producing language weights w_l that sum to one.

3.2 Team Composition Algorithm

Let $D = \{d_1, \ldots, d_n\}$ be the developers and $MS = \{ms_1, \ldots, ms_k\}$ the microservices. A configuration $c = \{T_1, \ldots, T_k\}$ assigns a team T_i to each microservice ms_i. The generator first guarantees at least two developers per microservice by shuffling the candidate pool and assigning the first $2k$ developers across services. Remaining developers are then assigned randomly. When the candidate pool is smaller than required, the implementation allows developer overlap across multiple microservice teams to keep the configuration feasible. Duplicate configurations are removed using a canonical key based on sorted team memberships. Candidate configurations are evaluated by the scores below, and local search with random restarts improves assignments by swapping developers between teams; a swap is accepted when it improves the tuple $(\text{minScore}, \text{meanScore}, -\text{stdScore})$.

Traits are encoded as LOW=0, MEDIUM=1, HIGH=2. For each team, the personality score is:

$$Score_{pers}(T_i) = 3\mu_C(T_i) - \sigma_C^2(T_i) + 2.5\mu_A(T_i) - 0.5\sigma_A^2(T_i) + \min_A(T_i) - 2\mu_N(T_i) + \mu_E(T_i) + \mu_O(T_i).$$

The weighting reflects the relative support found in the literature: strong support for high average *Conscientiousness* and *Agreeableness*, moderate support for low variance in *Conscientiousness* and *Agreeableness*, explicit reward for minimum *Agreeableness*, negative weight for *Neuroticism*, and smaller positive weights for *Extraversion* and *Openness*. The support column indicates the strength of the literature evidence used to guide weighting; it is not a multiplicative factor in the formula. We normalize the raw score to $[0, 1]$ as

$$\widehat{Score}_{pers}(T_i) = \frac{Score_{pers}(T_i) - (-4)}{17 - (-4)}.$$

Technical alignment is computed as the best language coverage inside a team:

$$Score_{lang}(T_i, ms_i) = \sum_{l \in L(ms_i)} w_l \cdot \max_{d \in T_i} skill(d, l),$$

Table 1. How literature-supported rules are encoded in $Score_{pers}$. Support indicates the strength of the literature evidence used to guide weighting; it is not a multiplicative factor in the formula.

Rule	Encoded term	Support
High average *Conscientiousness*	$+3\mu_C$	5/5
Low variance in *Conscientiousness*	$-\sigma_C^2$	4/5
High average *Agreeableness*	$+2.5\mu_A$	5/5
Low variance in *Agreeableness*	$-0.5\sigma_A^2$	3/5
High minimum *Agreeableness*	$+\min_A$	4/5
Low average *Neuroticism*	$-2\mu_N$	3/5
Moderate/high *Extraversion* and *Openness*	$+\mu_E, +\mu_O$	3/5

Table 2. Baseline and generated-configuration results. "mean pers." is the normalized mean personality score; ranges summarize the ten generated configurations.

Project	Baseline mean pers.	Baseline min. pers.	Gen. team size	Gen. mean pers. range	Mean lang range	Mean comb. range
HMDA	0.505	0.436	2–4	0.485–0.547	0.004–0.012	0.489–0.554
Taskcluster	0.520	0.395	2–4	0.429–0.566	0.620–0.662	1.066–1.227
Spinnaker	0.512	0.455	2–7	0.520–0.550	0.333–0.418	0.872–0.964

and the combined score is

$$Score_{comb}(T_i) = \widehat{Score}_{pers}(T_i) + \lambda Score_{lang}(T_i, ms_i).$$

At configuration level, we report the minimum, mean, and standard deviation of team scores.

4 Results

Table 2 summarizes the main results. In **HMDA-Platform**, most generated configurations improve the normalized mean personality score over the baseline of 0.505, with the best reaching 0.547. Because baseline and generated teams have comparable size (2–4 developers per microservice), the improvement is attributable mainly to developer reassignment rather than to a team-size artifact. The best optimized minimum personality score is 0.441, slightly above the baseline minimum of 0.436. Language scores are close to zero (0.004–0.012), so the combined score mostly mirrors personality composition; this suggests a homogeneous technology stack rather than poor technical fit.

In **Taskcluster**, only the top configurations exceed the baseline mean of 0.520, with the best reaching 0.566. The project has the most constrained structure: original teams contain only 1–3 developers and the generated configurations use 2–4. Hence, improvements depend on specific reassignment decisions rather than on averaging. The low baseline minimum (0.395) confirms that very small

service teams are highly sensitive to individual profiles; the observed spread in generated mean scores (0.429–0.566) shows that the algorithm is useful for surfacing viable high-scoring alternatives even when the search space is narrow.

In **Spinnaker**, all generated configurations improve the mean personality score over the baseline (0.512 to 0.520–0.550), but the baseline minimum (0.455) remains slightly above the best optimized minimum available in the detailed results (0.453). This reflects the effect of large-team averaging: the original project has up to 44 contributors per service, which naturally stabilizes scores, while generated teams are much smaller (2–7 developers). Language alignment differentiates Spinnaker configurations more than HMDA, suggesting that technical alignment becomes informative when the ecosystem is technologically diverse.

5 Discussion and Threats to Validity

The results answer **RQ1** positively at the level targeted by the paper: the algorithm identifies candidate configurations with higher average personality composition scores than the existing configuration in each project. This contribution is a decision-support mechanism for comparing alternative assignments under explicit personality and technical criteria. The result is strongest in HMDA, where baseline and generated team sizes are comparable, and more constrained in Taskcluster and Spinnaker, where structure respectively limits redistribution or creates strong averaging effects.

For **RQ2**, project structure strongly moderates the results. HMDA provides the cleanest case because baseline and generated team sizes are similar. Taskcluster shows that very small teams leave little room for redistribution. Spinnaker shows the opposite issue: large existing teams produce stable baseline scores through averaging, making optimized small-team configurations more variable. For **RQ3**, language alignment is project-dependent: it is negligible in HMDA, more discriminative in Spinnaker, and constrained by the small-team setting in Taskcluster.

Construct validity is mainly affected by indirect measurement. We mitigate personality-inference instability by combining multiple models, preprocessing variants, stability-based model selection, and majority voting; nevertheless, GitHub text primarily represents professional communication, and direct survey validation would further strengthen future studies. Language expertise is approximated through contribution proportions, which captures observed technical exposure but not necessarily depth of proficiency. Internal validity is limited by randomized generation and local search: the reported configurations are locally optimized candidates rather than guaranteed optima.

6 Conclusions

We proposed an algorithmic approach for composing microservice teams by combining Big Five personality composition and programming-language expertise.

Across three OSS microservice projects, the approach identifies candidate configurations with improved average personality composition scores and explains when such improvements are feasible: they are clearest when teams are small enough for individual profiles to matter but flexible enough to allow reassignment. The study therefore turns personality-aware team composition from a conceptual proposal into an operational, auditable optimization procedure. Future work can extend this procedure by validating inferred traits against self-reported measures, relating composition scores to observable development outcomes, and adding explicit objectives for worst-case team quality.

References

1. Barrick, M.R., Stewart, G.L., Neubert, M.J., Mount, M.K.: Relating member ability and personality to work-team processes and team effectiveness. Journal of Applied Psychology 83(3), 377–391 (1998)
2. Bazelli, B., Hindle, A., Stroulia, E.: On the personality traits of StackOverflow users. In: Proc. IEEE International Conference on Software Maintenance (ICSM), pp. 460–463. IEEE (2013). https://doi.org/10.1109/ICSM.2013.72
3. Bell, S.T.: Deep-level composition variables as predictors of team performance: A meta-analysis. Journal of Applied Psychology 92(3), 595–615 (2007)
4. Calefato, F., Lanubile, F.: Using personality detection tools for software engineering research: How far can we go? ACM TOSEM 31(3), 1–48 (2022). https://doi.org/10.1145/3491039
5. Calefato, F., Lanubile, F., Vasilescu, B.: A large-scale, in-depth analysis of developers' personalities in the Apache ecosystem. Information and Software Technology 114, 1–20 (2019). https://doi.org/10.1016/j.infsof.2019.05.012
6. Costa, P.T., Jr., McCrae, R.R.: Revised NEO Personality Inventory (NEO-PI-R) and NEO Five-Factor Inventory (NEO-FFI) Professional Manual. Psychological Assessment Resources, Odessa, FL (1992)
7. Fowler, M., Lewis, J.: Microservices: A definition of this new architectural term. https://martinfowler.com/articles/microservices.html (2014), accessed 2025-05-01
8. Goldberg, L.R.: An alternative "description of personality": The Big-Five factor structure. Journal of Personality and Social Psychology 59(6), 1216–1229 (1990). https://doi.org/10.1037/0022-3514.59.6.1216
9. Hurtz, G.M., Donovan, J.J.: Personality and job performance: The Big Five revisited. Journal of Applied Psychology 85(6), 869–879 (2000)
10. Li, X., Calefato, F., Lenarduzzi, V., Taibi, D.: Toward collaboration optimization in microservice projects based on developer personalities. In: Proc. 21st IEEE International Conference on Software Architecture – Companion (ICSA-C), pp. 95–99. IEEE (2024). https://doi.org/10.1109/ICSA-C63560.2024.00024
11. Mathieu, J.E., Maynard, M.T., Rapp, T., Gilson, L.: Team effectiveness 1997–2007: A review of recent advancements and a glimpse into the future. Journal of Management 34(3), 410–476 (2008)
12. Neuman, G.A., Wagner, S.H., Christiansen, N.D.: The relationship between work-team personality composition and job performance. Group & Organization Management 24(1), 28–45 (1999)

13. Peeters, M.A.G., van Tuijl, H.F.J.M., Rutte, C.G., Reymen, I.M.M.J.: Personality and team performance: A meta-analysis. European Journal of Personality 20, 377–396 (2006)
14. Prewett, M.S., Brown, M.I., Goswami, A., Christiansen, N.D.: Effects of team personality composition on member performance. Group & Organization Management 43(2), 316–348 (2018)
15. Prewett, M.S., Walvoord, A.A.G., Stilson, F.R.B., Rossi, M.E., Brannick, M.T.: The team personality–team performance relationship revisited. Human Performance 22, 273–296 (2009)
16. Rigby, P.C., Hassan, A.E.: What can OSS mailing lists tell us? A preliminary psychometric text analysis of the Apache developer mailing list. In: Proc. Fourth International Workshop on Mining Software Repositories (MSR'07). IEEE (2007). https://doi.org/10.1109/MSR.2007.35