

AI Merge Integrity Assessment

PUBLIC METHODOLOGY SAMPLE

Repository	microsoft/kiota (public)
Engine	Merge-Proof ohcaygo/merge-proof at 44a8f715eafaad4178743aa2c46a8bf448ff0a54
Prepared	3 September 2026
Classification	Demonstration. Not a customer engagement.

Microsoft is not a customer. This assessment is not commissioned by, affiliated with, or approved by Microsoft. It uses only public Git history and public GitHub pages. Findings are **evidence classifications**, not defect allegations. NOT_PROVEN does not mean the code is wrong.

2. Executive summary

What we measured. Across squash-style merges on microsoft/kiota whose Git trailers identified an AI coding agent, did the repository contain evidence that the candidate was checked against the base it actually landed on?

What we found. The collector saw 77 such merges (67 labeled Copilot, 10 labeled Claude). **77 analyzed, 0 unresolvable. 75 VERIFIED (97.4%), 2 NOT_PROVEN (2.6%).** Both NOT_PROVEN rows were independently re-run against the shipped analyzer; verdicts match the published study JSON.

After **manual review of the two hits** (required; path regex is not semantic truth):

1. **PR #8093** — BASE_DRIFT_UNVERIFIED. Base advanced 2 commits; the only overlapping file is CHANGELOG.md. Public timeline shows the PR entered the merge queue, was removed, then squash-merged by a human. Missing evidence is a validation of the **combined** candidate+new-base state. This does **not** establish that the OpenAPI.NET bump was incorrect. 2. **PR #7981** — PROTECTED_BOUNDARY / policy. The path is .github/policies/branch-protection.yml. That matches a filename regex for “policy”; it is **GitHub branch-protection GitOps**, not access-control / RLS. Analyst classification: **not a protected-boundary concern of the kind the category describes**. The PR’s public description is enabling a merge queue and turning off strict status checks.

What it means. On the population this method can see, kiota’s squash-merged agent-trailer PRs almost always have no implemented evidence-gap. The residual rate is low. One of two tool hits is a known class of regex misfire.

What it does not mean. It does not measure bugs, security, CI-on-final-head, human approval, or Microsoft’s internal process. It does not measure rebase or regular merge commits (3,040 of 4,000 first-parent commits in the scan window have two parents and are **invisible** to this collector). It does not prove Copilot “wrote” 67 PRs — trailers include Copilot Autofix on human-authored work.

Recommendation. Do nothing as a default for this public repository, with two optional control notes: (a) treat .github/policies/** as a documented exclusion if this method is reused; (b) if merge-queue completeness matters, inspect **why** #8093 left the queue — that question is outside Merge-Proof.

3. Scope

Item	Value
Repository	https://github.com/microsoft/kiota
Default branch	main
Engine SHA	44a8f715eafaad4178743aa2c46a8bf448ff0a54
Population	Merged PRs reconstructed as single-parent squash commits on first-parent history whose author/committer/Co-authored-by text matches shipped agent patterns, with a (#N) subject
Detected	77 (agentMergesFound = 77)
Cap	Collector - -max-prs 80; cap did not bind
Scan window	First-parent, - -max-count=4000
Squash-commit date range of the 77	30 Oct 2025 – 31 Aug 2026 (committer dates)
PR number range	#6416 – #8110 (numbers are not time order)
Exclusions	Dependabot/Renovate not in agent list. Generated/lockfile paths excluded from overlap. ci-deploy paths advisory only
Not in scope	Private Microsoft repos, internal teams, non-squash merges, CI logs, review API, ruleset admin UI

Inclusion is **purposive**: kiota was in the G1 study list because it was expected to merge agent-authored PRs. Rates here are not GitHub-wide rates.

4. Methodology (plain English)

Merge-Proof asks one question: **did anything actually check this change against the tree it merged into?** A clean Git merge is not that evidence.

The historical collector (network: GitHub clone/fetch; **not** the core analyzer) blobless-clones the repo, walks first-parent history, keeps squash commits with agent trailers and a PR number, fetches refs/pull/N/head, and runs the same src/analyze.js that ships in the package.

The **core analyzer** is local Git only: no model, no telemetry requirement.

Checks implemented in this release:

- BASE_DRIFT_UNVERIFIED — base moved, and overlapping files exist.
- PROTECTED_BOUNDARY — candidate paths match blocking regex categories (migration, schema, auth, secrets, billing, policy).

Not implemented (so VERIFIED is not a claim about them): CI on final head, human approval, remote durability, scope vs declaration.

For a paid engagement the intended model is: **the customer runs collection with their access and shares JSON**. This sample used public GitHub because the repo is public.

Technical reproduction is in the appendix.

5. Coverage and data integrity

This section is the denominator. Read it before the rate.

Layer	Count	Notes
First-parent commits scanned	4,000	<code>git log --first-parent origin/main --max-count=4000</code> on the verification clone
Multi-parent (regular merges)	3,040	Invisible to squash reconstruction
Single-parent	960	Candidate pool for squash+trailer detection
Agent-trailer squash merges found	77	<code>agentMergesFound</code>
Analyzed	77	All 77 squash SHAs present on the clone (0 missing)
Unresolvable PR heads	0	
Coverage of *detected* population	77/77 = 100%	
Coverage of *all merges*	Unknown, and likely far below 100%	Merge-commit era dominates the 4,000-commit window

Merge-strategy coverage. Kiota historically used merge commits. A public PR (#7981, 23 Jul 2026) states a ruleset was created with `merge_method: SQUASH`. Of the 77, **60** have squash dates before 23 Jul 2026 and **17** on/after. The method is more aligned with post-queue squash policy than with earlier merge-commit history.

Agent identification. Patterns in `study/collect.js`: Copilot, Claude, Cursor, Devin, Codex, Aider, Sweep, OpenHands, Cody, Jules — in author, email, or Co-authored-by. Human PRs with Copilot Autofix trailers count. Agents that leave no trailer are missed.

****What we did *not* resolve. Ruleset admin state (merge queue on/off today) is not fully visible in-tree. CI conclusions per SHA are not**** read.

6. Merge-integrity results

Headline metric **only with the denominator above**.

Verdict	Count	Share of 77
VERIFIED	75	97.4%
NOT_PROVEN	2	2.6%
FAIL	0	0%
Unresolvable (excluded from rates)	0	—

Integrity rate (implemented checks only): 75/77 VERIFIED.

Independent verification (this sample): both NOT_PROVEN SHA pairs were re-analyzed with shipped `analyze()` on a blobless clone. Verdicts, `baseAdvanceCommits`, and finding IDs match `study-data.json`. File lists below come from that re-run, not from memory.

Agent labels among the 77: Copilot 67, Claude 10.

Base advanced under the candidate (`baseAdvanceCommits > 0`) in **12** of 77; overlapping files in **1** of 77. Drift without overlap does not produce NOT_PROVEN.

7. Evidence-gap distribution

Finding	PRs	Rate of 77	After analyst review
BASE_DRIFT_UNVERIFIED	1	1.3%	Stands as an evidence gap; overlap is <code>CHANGELOG.md</code> only
PROTECTED_BOUNDARY	1	1.3%	Does not stand as access-policy/RLS; path regex misfire
Other implemented gaps	0	0%	—

8. Protected-boundary exposure

Categories with **zero** hits in this population: auth, schema, migration, secrets, billing.

Category	Tool hits	Manual review
policy	1	.github/policies/branch-protection.yml — GitHub Policy Service YAML, not application authorization. Do not treat as semantic policy/RLS.

Observed limitation (not a feature request): the policy regex matches policies/ at any path, including .github/policies/.

9. Concentration

No internal Microsoft teams are inferred.

- **By finding type:** 1 changelog overlap, 1 GitHub-policy path.
- **By month of squash (77):** 2025-10: 1; 2025-11: 1; 2026-01: 2; 2026-02: 3; 2026-03: 4; 2026-04: 15; 2026-05: 6; 2026-06: 5; 2026-07: 24; 2026-08: 16.
- **Post-MQ-enable (squash date $\geq$ 23 Jul 2026):** 17 PRs, of which 1 NOT_PROVEN (#8093). #7981 is the MQ-enable change itself (22–23 Jul).

10. Top findings (SHA-level)

Finding A — PR #8093 — BASE_DRIFT_UNVERIFIED

Public PR	https://github.com/microsoft/kiota/pull/8093
Title (public)	chore(deps): update OpenAPI.NET version
GitHub author	gavinbarron
Collector agent label	copilot
Squash	6bb3e640237e0cae33d2672107d0b8498369d094
Base at merge (first parent)	73ab4489e15820fbc63108a373a9b11981445ff6 (2026-08-28T18:39:39Z)
Candidate head	9554d4f2ed0388f66cd982988c035756f9b05ad8 (2026-08-28T18:55:09Z)
Fork point	341844021fc041361f5a0e98fe3c4d0b50b22517
Merged (public)	2026-08-28

Observed fact. Candidate changed 5 files: CHANGELOG.md, src/Kiota.Builder/Kiota.Builder.csproj, src/Kiota.Builder/OpenApiDocumentDownloadService.cs, and two test files. Base advanced 2 commits; 1 overlapping file: CHANGELOG.md.

Classification. NOT_PROVEN / BASE_DRIFT_UNVERIFIED — implemented rule fired.

Why classified. Overlap is non-empty. The analyzer does not exclude changelogs.

What it does not prove. It does not prove the dependency bump is wrong, that CI failed, or that a human skipped review. Public timeline includes reviews and a coverage comment on head 9554d4f. Merge-Proof does not read those as evidence of *combined-state* validation.

Public control note (not from the analyzer). Timeline includes added_to_merge_queue then github-merge-queue[bot] removed_from_merge_queue then a human squash-merge. Cause of removal is **not established**.

Possible control response. Rebase/re-queue so changelog overlap is validated with the new base; or exclude CHANGELOG.md from overlap if that is accepted mechanically. Neither is required by the rate.

Finding B — PR #7981 — PROTECTED_BOUNDARY (policy) — analyst override

Public PR	https://github.com/microsoft/kiota/pull/7981
Title (public)	Enable merge queue on main and drop strict status checks
GitHub author	gavinbarron
Collector agent label	copilot
Squash	591d97e4d20eab8f96b3675064b11b78b9376daa
Base	21dc6a2bb62d21bf83923d717d15dd33471ca3e4 (fork point = base; advance 0)
Candidate head	e5fb44d44b4f4760ee9c41525cc0c6751be06bb1
Merged (public)	2026-07-23

Observed fact. One file: .github/policies/branch-protection.yml. Diff sets requiresStrictStatusChecks: false. PR body (public) states a **repository ruleset** merge-queue-default-branch was created via API with squash, because Policy Service schema cannot express merge queue.

Classification by tool. NOT_PROVEN / PROTECTED_BOUNDARY / policy.

Classification by analyst. Regex misfire. /(^\|\/)(policies\|policy|rls|rbac|permissions?)(\|\/\$)/i matches .github/policies/. This file is branch-protection GitOps, not application policy/RLS.

What it does not prove. It does not prove a permissions regression. It does not prove the merge queue ruleset exists today (that lives in repo settings, not fully in this file).

Possible control response. If reusing Merge-Proof here, document .github/policies/ as an **exclusion** or** accept the false semantic. Do not treat this hit as a reason to change kiota process.

11. Control observations (public in-tree / public PRs only)

Do not read this as a description of Microsoft's private process.

Visible in public files / pages:

- CODEOWNERS: * @microsoft/kiota-write
- .github/policies/branch-protection.yml on main (fetched 3 Sep 2026): PR required; **1** approving review; **CODEOWNERS review required**; conversation resolution required; required checks named: license/ccla, Analyze (csharp), dotnet-build, check-idempotency-tests, check-integration-tests, build_extension; **requiresStrictStatusChecks: false**
- PR #7981 public text: merge queue via **ruleset**, squash method, Policy Service cannot store MQ
- PR #8093 public timeline: merge queue add then remove, then human merge

Unknown without admin/API: whether the ruleset is still active, queue depth, required merge_group checks.

12. Recommendations (proportional)

Ranked, for **this public sample only**:

1. **NO CHANGE / DO NOTHING** — default. 97.4% VERIFIED on the visible population; remaining tool hits do not justify a new gate. 2. **FURTHER INTERNAL REVIEW** (optional) — only if queue completeness is a goal: why #8093 left the queue. Merge-Proof cannot answer that. 3. **OTHER EXISTING CONTROL** — keep CODEOWNERS + named required checks already declared. No evidence they are absent. 4. **MERGE QUEUE** — already the direction of #7981. This sample does not justify *introducing* a queue; it notes public intent to use one. 5. **REQUIRED CHECK** — already listed. Adding Merge-Proof as required is **not** justified by 2.6% with one misfire. 6. **MERGE-PROOF GATE** (-fail-on not-proven) — **not recommended** here without first excluding .github/policies/** and deciding changelog overlap policy; otherwise it would have blocked #7981, a process improvement.

13. Do-nothing option

The rational decision is do nothing when:

- the visible NOT_PROVEN rate is low;
- remaining hits are changelog mechanics or path-regex misfires;
- public controls already include reviews, CODEOWNERS, and named checks;
- a merge queue is already being adopted.

Paying for a continuous gate, a merge-queue vendor, or a \$7,500 follow-on **on the strength of this kiota public sample alone** is not supported. A private team with different volume, no queue, and protected application paths could reach a different recommendation. That is the point of a scoped pilot, not of this demo.

14. Limitations (prominent)

- Public data only. Microsoft is not a customer.
- Squash-subject reconstruction; **rebase and regular merges missed** (3,040/4,000 first-parent commits multi-parent).
- Agent trailers, including Copilot Autofix on human PRs.
- PR heads that GitHub has garbage-collected would be unresolvable (0 here).
- No semantic bug, security, vulnerability, or correctness analysis.
- No human-approval verification.
- No CI-final-head proof.
- Path categories are regex, not meaning — demonstrated by Finding B.
- Changelog overlap is treated like any other file — demonstrated by Finding A.
- Core analyzer is local Git; **the collector** clones/fetches GitHub.
- VERIFIED means “implemented checks did not fire,” not “safe.”

15. Technical appendix

Reproduction (existing tooling only)

```
# Engine at frozen SHA
git clone https://github.com/ohcaygo/merge-proof.git
git -C merge-proof checkout 44a8f715eafaad4178743aa2c46a8bf448ff0a54

# Published corpus (already in repo)
# study-data.json → microsoft/kiota: 77 rows, 75 VERIFIED, 2 NOT_PROVEN, 0 unresolvable

# Independent re-analyze (this sample)
git clone --filter=blob:none --no-checkout https://github.com/microsoft/kiota.git
git fetch origin +refs/pull/8093/head +refs/pull/7981/head
# analyze({ base: 73ab4489..., head: 9554d4f2... }) → NOT_PROVEN BASE_DRIFT_UNVERIFIED overlap CHANGELOG.md
# analyze({ base: 21dc6a2b..., head: e5fb44d4... }) → NOT_PROVEN PROTECTED_BOUNDARY policy .github/policies/branch-
protection.yml
```

Collector (historical; uses network): node study/collect.js --repos <file-with-microsoft/kiota> --max-prs 80 --max-commits 4000.

NOT_PROVEN list

PR	Agent label	Squash	Head	Finding
8093	copilot	6bb3e640237e0cae33d2672107d0b8498369d094	9554d4f2ed0388f66cd982988c035756f9b05ad8	BASE_DRIFT_UNVERIFIED / CHANGELOG.md
7981	copilot	591d97e4d20eab8f96b3675064b11b78b9376daa	e5fb44d44b4f4760ee9c41525cc0c6751be06bb1	PROTECTED_BOUNDARY / .github/policies/branch-protection.yml

Claims removed for lack of evidence

- Any implication that kiota or Microsoft shipped defective, untested, or unsafe code.
- Any SOC 2 / SOX / audit-failure claim.
- Treating 24.1% G1-study headline as kiota’s rate (kiota is 2.6% on this population).
- Treating 67 “copilot” rows as fully agent-authored PRs.
- Asserting the merge-queue ruleset is in force on 3 Sep 2026.

- Asserting CI did or did not run on the combined #8093 tree.
- Network-free claims about the **collector**.

Observed limitations (no engineering tasked)

- policy regex vs .github/policies/.
- CHANGelog.md in overlap.
- Squash-only collector vs merge-commit history.