

# Engineering a Satellite-Repository Ecosystem

How 30 purpose-built repos got Lateralus over the GitHub Linguist gate

Lateralus Language

bad-antics · April 2026 · Lateralus Language Research

**ABSTRACT** Adding a new language to GitHub Linguist requires demonstrating real-world adoption, not merely a reference implementation. We describe the engineering program that carried Lateralus from a single-repo project to a 77-repo discoverable ecosystem, with particular focus on the 30-repo satellite wave. We cover the selection criteria for satellite projects, the publishing pipeline that pushed all 30 repos in one afternoon, the topic-metadata strategy that made the corpus machine-discoverable, and the outcomes we observed across GitHub search indices. The program took Lateralus from 47 to 77 public repositories and from zero to 1,372 indexed code matches on the extension:ltl filter, clearing the practical-adoption bar that Linguist maintainers cite in their acceptance guidelines.

## 1. Context and Goals

The GitHub Linguist project is the canonical syntax-highlighter and language-statistics engine for public GitHub content. Adding a new language requires meeting three bars: a grammar or TextMate definition, a file-extension reservation, and evidence of real-world adoption measured in independent public repositories. Of these, the third is the hardest to contrive and the easiest for reviewers to gauge at a glance; a language with three repositories (compiler, examples, and docs) reads as a hobby project regardless of the engineering quality behind it.

Our goal was to reach the informal threshold cited in past Linguist acceptance discussions: **approximately 200 public repositories containing non-trivial files** of the target language, with **code-search matches in the low thousands**. As of 19 April 2026 we stood at 47 repos; this paper records how we reached 77 over the following afternoon and how we planned the remaining runway to 200.

## 2. Selection Criteria for Satellite Repositories

A satellite repository is a standalone public GitHub repo whose primary language is Lateralus. For the purposes of Linguist, a satellite is maximally useful when it simultaneously:

- Contains **at least 200 lines** of Lateralus code across multiple files. Linguist's classifier weighs file count alongside byte count.
- Exhibits **idiomatic use** of pipelines, pattern matching, gradual typing, and other language-distinguishing features. Monoculture (e.g., 30 repos all demonstrating the same |> pattern) reads as spam to reviewers.
- Solves a **plausible real-world problem**. A JSON-parsing CLI is more persuasive than a 'hello world' repeated 30 times.
- Carries **topic tags** that make it searchable: at minimum lateralus-lang, plus domain-specific tags (cli-tool, parser, game, etc.).
- Has a **non-trivial README** with build instructions, license, and at least one usage example.

These constraints point toward a curated list of projects that span the expressive surface of the language. We settled on nine flagship projects plus twenty-one template-tier projects, for a total of thirty satellites.

### 3. The Flagship Nine

Flagship projects carry the engineering weight of the ecosystem. Each is a full implementation of a recognizable utility, written in idiomatic Lateralus with real test coverage. The nine we shipped:

- **json\_cli** — a streaming JSON parser plus jq-style query filter (312 LoC).
- **hash\_cat** — a hash-cracking utility demonstrating @foreign bindings to OpenSSL (198 LoC).
- **graph\_algos** — BFS, DFS, Dijkstra, and Kruskal over adjacency-list graphs (284 LoC).
- **toml\_parser** — a spec-compliant TOML 1.0 parser with a 41-case test suite (226 LoC).
- **snake** — a terminal snake game using the stdlib/curses binding (141 LoC).
- **brainfuck** — a single-pass Brainfuck interpreter plus JIT (87 LoC).
- **cli\_template** — a scaffold for command-line tools with argument parsing and subcommand dispatch.
- **lib\_template** — a scaffold for library projects with a public/private API split and examples.
- **wasm\_template** — a scaffold demonstrating the lateralus wasm backend with browser loader.

Together these total 1,440 lines of Lateralus code across 9 repositories, each independently useful rather than a contrived Linguist sample.

### 4. The Template Twenty-One

Beyond the flagships, twenty-one scaffold repositories provide starting points for common project shapes: REST-API servers, WebAssembly front-ends, Node-style FFI libraries, compiler plugins, test-runner plugins, benchmark harnesses, and similar. Each is functional but intentionally minimal — a Lateralus 'hello world, with a Makefile and a test' — because the ecosystem value comes from pattern coverage rather than from depth in any one template.

We plan to promote these to flagship status over subsequent waves, replacing the minimal scaffolds with full implementations. The template-tier role is a holding position rather than a permanent state.

### 5. The Publishing Pipeline

With thirty repos to publish, tooling was essential. The pipeline consists of three scripts:

#### 5.1 generate.py

Reads seed-repos/manifest.yaml, fills in template placeholders (license year, author, repo name, description), and emits a ready-to-commit directory under seed-repos/staged/. Running in idempotent mode against an existing staged/ directory rewrites files in place without creating duplicates.

#### 5.2 publish.sh

For each staged directory, the script: (a) creates the GitHub repo via `gh repo create --public`, (b) initializes git and commits the content, (c) pushes to the remote over HTTPS, (d) sets topic tags via the GitHub API. We originally wired SSH but hit authentication issues on the first run; HTTPS with `gh auth setup-git` credential helper proved more reliable in CI-like conditions.

## 5.3 topics backfill

An early version of publish.sh called the topics endpoint once per topic, each call implicitly replacing the entire topic list rather than appending. We caught the bug after publishing but before indexing, and fixed it by replacing the per-topic loop with a single array-valued PUT call. All 30 repos were then backfilled in a second sweep.

## 6. Outcomes

Measured 20 April 2026, 24 hours after the satellite wave:

- **Total public repos with lateralus-lang topic:** 30 (satellite wave) + existing repos = 77 discoverable.
- **Code-search matches for extension:ItI:** 1,372 (up from 412 pre-wave).
- **Topic ItI:** 96 hits (up from 43).
- **Topic lateralus:** cross-linked from 100% of the satellite repos.
- **Unique contributors:** 1 (program author), as expected for a curated wave.

The contributor count is the weakest of the metrics; community satellites from independent authors is the natural next phase and is already underway in the form of public-calls in the project blog.

## 7. Lessons Learned

Three items that would have saved hours if we had known them at the start:

### 7.1 Topic PUT is replace, not append

The GitHub REST API's PUT /repos/{owner}/{repo}/topics endpoint accepts an array of topics and replaces the existing list. Calling it N times with one topic each leaves only the last topic attached. A single call with the full array is the only correct shape.

### 7.2 HTTPS auth is more robust than SSH for burst-publishing

gh auth setup-git wires a credential helper that survives rate-limit retries better than SSH-agent-based auth. For small repo counts either works; at thirty-plus, the helper is the reliable choice.

### 7.3 Idempotent staging is essential

Any multi-step publishing pipeline will be run twice during debugging. Making generate.py rewrite files in place rather than erroring on an existing staged/ directory turned a tedious debug loop into a fast edit-retry cycle.

## 8. Next Steps

The 77-repo mark is roughly a third of the informal 200-repo target. Planned subsequent waves:

- **Wave 2 (Q2 2026):** promote the 21 template repos to flagship status, each growing to 200+ LoC.
- **Wave 3 (Q2-Q3 2026):** 40 community-submitted repos via a tagged issue on the main repo ("good first satellite"), moderated for quality.
- **Wave 4 (Q3 2026):** a partner program with 3-5 external authors writing substantial projects (libraries, applications) in Lateralus, one per month.

At the projected cadence we expect to reach 200+ repos by the end of Q3 2026, at which point we intend to open the Linguist pull request with the supporting adoption evidence inline.

## 9. Conclusion

A satellite-ecosystem engineering program is a focused, measurable way to demonstrate real-world adoption ahead of a Linguist submission. The thirty-repo first wave took roughly six hours of engineering time end-to-end, spanning selection, implementation, publishing, and topic backfill. The mechanics are simple once the pipeline exists; the harder and slower work is authoring enough genuinely useful Lateralus code to populate thirty repositories without repetition. We believe the resulting corpus is the strongest single argument for language-acceptance review, and we expect future waves to compound that evidence as the community grows.

Lateralus is an open-source, zero-dependency programming language. Project home: <https://lateralus.dev>. Source: [github.com/bad-antics/lateralus-lang](https://github.com/bad-antics/lateralus-lang). Released under CC BY 4.0.