

# Ahrefs Blog Pipeline — Ryan Law

## Summary

Контент-автоматизация Ahrefs на Claude Code: ~23 skill-файла стянутые двумя master skills ( `blog-pipeline` , `update-pipeline` ). От ключевика до готового черновика — 8–12 минут. На момент интервью: ~30 обновлений статей + 10–15 новых опубликовано.

Главная идея — не "лучший промпт для генерации статьи", а декомпозиция на много шагов (research, references, outline, product placement, draft, claims, preview, screenshots, formatting), где каждый шаг сохраняет промежуточный артефакт. Это даёт интроспекцию и точку для исправления, если что-то пошло не так.

## Key Details

### Архитектура

- Папка `blog-pipeline/` с ~23 skill-файлами, каждый — markdown с workflow
- В корне — `keyword-ideas.csv` (опционально, под content gap analysis через Ahrefs MCP)
- 2 master skill'a: `blog-pipeline` (новая статья) + `update-pipeline` (обновление существующей)
- Каждый skill можно дёрнуть отдельно (нужен только outline → запускаешь outline skill)
- Команда форкает репо под себя: voice profile свой, набор шагов свой

### 12 шагов (новая статья)

#	Шаг	Что делает
1	<b>Context trigger</b>	На входе можно дать произвольный текст: "покрой эту тему, упомяни этот продукт, ссылка на эту статью" — задаёт уникальный угол
2	<b>Research</b>	Ahrefs MCP: primary keyword + parent topic + longtail variations + questions report. Web fetch топ-выдачи → анализ заголовков, тем, gap'ов. Финал — research-report
3	<b>Ahrefs references</b>	Ищет уже опубликованные статьи Ahrefs по теме → выдёргивает релевантные секции, продуктовые use cases, anchor texts для внутренней перелинковки
4	<b>Outline</b>	BLUF principle, MECE-секции, hooks/key points/transitions/examples/таблицы — кости статьи

#	Шаг	Что делает
5	<b>Product annotation</b>	Отдельный шаг: проходит по outline, мечает где упомянуть Keyword Explorer / Site Audit / Rank Tracker / Content Explorer / Firehose. Master-список продуктов внутри skill
6	<b>Draft</b>	Пишет полный черновик. Внутренний style guide Ahrefs: PAS-формула во вступлении, inverted pyramid, "always explain what and why"
7	<b>Claims verify</b>	Сканирует черновик на утверждения требующие источник → ищет up-to-date research/stats → подставляет ссылки
8	<b>Citation / internal linking</b>	Внутренние ссылки на ahrefs.com, supporting external sources
9	<b>Screenshots</b>	Самое слабое место. Skill умеет генерить корректные URL Ahrefs-отчётов (отдельный коллега написал) → Ryan руками скриншотит. Ryan пробует автоматизировать через headless browser
10	<b>Preview</b>	Генерит styled HTML под скин Ahrefs blog — Ryan проверяет в браузере вместо чтения markdown
11	<b>Formatting for publish</b>	WordPress shortcodes, всё что нужно для CMS
12	<b>Output</b>	Готовый черновик на ревью

## Update pipeline (отдельный flow)

Для тысячи уже опубликованных статей. Делает 3 вещи:

1. **Outdated claims** — старые статьи/цифры → находит свежую версию → предлагает заменить
2. **Product feature gaps** — статья 8-летней давности не упоминает Firehose / AI Content Helper → рекомендация добавить
3. **Topic gaps** — смотрит SERP, ищет темы, которые покрывают конкуренты, но нет у Ahrefs → драфтит секцию для ревью

## На каких темах работает (граница применимости)

Ryan чётко проводит границу:

- **Подходит:** informational topics, "boring evergreen" темы где Ahrefs уже писал 100 раз в разных контекстах. AI хорошо research'ит и кросс-референсит существующий контент
- **НЕ подходит:** уникальные взгляды, материалы требующие мышления и опыта, контент с personal-experience hooks. Это они пишут руками

- Каждая статья на ревью у Ryan — он не публикует если не нашёл достаточно интересного

## Цикл проб и ошибок

- Сначала тестируют ПАРНО: с skill-файлом vs без skill-файла → если без skill результат такой же, skill упрощается/удаляется
- Skill-файлы постепенно distill'ятся до handful of важных принципов
- Ryan признаёт: "probably overengineered" — но структура помогает думать о процессе

## Related

- [ai-book-writing-dictation](#) — параллельный workflow Тима для написания книги через dictation
- [skills-system-ai-agents](#) — общая концепция skills как workflow (Max Mitcham / ColdIQ)
- [claude-code-content-engine](#) — похожий подход у ColdIQ для LinkedIn-контента
- [content-pipeline-7-steps](#) — 7-шаговый pipeline у ColdIQ (близкая декомпозиция)
- [content-foundation-docs](#) — voice profile / examples в skill-файлах как фундамент

## Sources

- [raw/articles/2026-04-28-ahrefs-ryan-law-ai-blog-pipeline](#)