

[• ZASTOSOWANIA](#)

Co SumizAI daje programistom?

Programiści i tak rozmawiają z modelami cały dzień, i wynik i tak znika cały dzień.

Konkretna strata to nie kod — kod jest w repozytorium. **To rozumowanie:** dlaczego to podejście, co zostało odrzucone, co naprawdę pokazał pomiar.

Aktualizacja: 2026-08-14 8 pytań 3 min czytania

Q. Co właściwie warto łapać?

Decyzje i ich alternatywy. Reszta jest albo w kodzie, albo niewarta zachowania.

Notatka mówiąca „używamy tej kolejki” jest warta prawie nic — mówi to konfiguracja. Notatka mówiąca „wybraliśmy tę kolejkę zamiast tamtej, bo semantyka ponownego dostarczania ważyła więcej niż przepustowość, a oto pomiar” jest warta bardzo dużo i dokładnie tego nie zapisuje żaden artefakt w repozytorium.

Druga kategoria to sesje debugowania zakończone wnioskiem. Poprawka to commit; powód, dla którego błąd był możliwy, to notatka.

Q. Dlaczego zasada wyłącznego dopisywania ma tu znaczenie?

Bo twierdzenia inżynierskie się starzeją, a Ty musisz wiedzieć kiedy.

Każda notatka trzyma kompletne oryginalne pytanie i odpowiedź, i nic późniejszego ich nie zmienia ani nie usuwa. Kiedy scalona notatka mówi, że podejście się nie skaluje, możesz przeczytać oryginalną wymianę i odkryć, że chodziło o konkretną wersję, przy konkretnym obciążeniu, w konkretnej konfiguracji.

Samo podsumowanie tego nie powie, a podsumowanie, którego nie da się zaudytować, jest sposobem, w jaki zespoły dorabiają się mądrości ludowej, której nikt nie potrafi obronić.

Q. Jak to pasuje obok repozytorium?

Bezpośrednio. Vault to katalog plików `.md` ze względnymi odnośnikami i wygenerowanym indeksem `base.md`.

Zacommituj go. Diffy są czytelne, review działa, blame działa, a notatki podróżują razem z kodem. Nic w tym formacie nie walczy z kontrolą wersji, bo to format, dla którego kontrola wersji powstała.

Oznacza to też, że działają Twoje dotychczasowe narzędzia — grep, ripgrep, podgląd Markdowna w edytorze, generator strony statycznej, jeśli chcesz wewnętrzny podręcznik.

Q. Jak wygląda dobry układ vaulta w pracy inżynierskiej?

Jeden vault na system albo na długo żyjącą dziedzinę, nie na ticket.

Użyteczne jest pytanie o granicę: kiedy pytam o to, czy model powinien widzieć tamto? Dla usługi, którą utrzymujesz — tak; decyzje o jej składowaniu, API i trybach awarii należą do siebie. Dla niepowiązanego produktu — nie.

Drzewo rozdziałów niesie wtedy strukturę wewnętrzną i schodzi na sześć poziomów, jeśli temat tego wymaga. Tickety nie są rozdziałami; podsystemy są.

Q. Czy mogę wskazać model lokalny?

Tak. Ollama jest jednym z siedmiu obsługiwanych dostawców, a własny serwer drugim — działa wszystko, co mówi protokołem OpenAI, bo sześciu z siedmiu dostawców dzieli jednego klienta.

Dla bazy kodu, której nie wolno nigdzie wysyłać, to jest konfiguracja, która w ogóle czyni aplikację używalną. Nic nie opuszcza maszyny.

Jeden szczegół operacyjny, jeśli uruchamiasz aplikację w kontenerze: adres w rodzaju `localhost` dla dostawcy na hoście jest przepisywany na bramę hosta, bo własny udokumentowany adres Ollamy jest z wnętrza kontenera nieosiągalny, a ta awaria jest myląca.

Q. A kod, który pisze model?

Trzymaj go w repozytorium. SumizAI nie jest menedżerem snippetów, a notatka to kiepski dom dla źródeł, które mają się kompilować.

W notatce należy się rozumowanie wokół kodu: na czym polega podejście, co zakłada, co próbowano najpierw. Fragmenty w notatce są w porządku jako ilustracja i fatalne jako kopia kanoniczna.

Granica jest mniej więcej taka: jeśli ma się uruchamiać, idzie do repozytorium; jeśli wyjaśnia, dlaczego to, co się uruchamia, wygląda tak, a nie inaczej — idzie do vaulta.

Q. Czy pomaga przy wdrażaniu nowych osób?

W ograniczony, ale realny sposób. Nie ma vaultów współdzielonych, więc to nie jest zespołowa baza wiedzy — ale eksporty są artefaktami do przekazania.

Vault eksportuje się jako Markdown, PDF albo prezentację w kolejności spisu treści. „Oto rozumowanie stojące za tą usługą, uporządkowane, czterdzieści stron” to znacznie lepsze przekazanie niż katalog linków.

Jeśli chcesz naprawdę współdzielonego, wersjonowanego źródła prawdy o tym, co system ma robić, to inne narzędzie z innym zadaniem.

Q. Gdzie zawodzi w zastosowaniach inżynierskich?

Brak synchronizacji między urządzeniami, więc laptop i desktop to osobne biblioteki.

Brak vaultów współdzielonych, więc jest to narzędzie osobiste używane wewnątrz zespołu, a nie narzędzie zespołowe.

Brak wyszukiwania semantycznego — wyszukiwanie jest leksykalne z dopasowaniem trigramowym, a ocena semantyczna dzieje się w modelu czytającym krótką listę. W praktyce jest to szybkie i wystarczające, ale jeśli spodziewałeś się embeddingów, nie ma ich.