BudujemyAI_

E-BOOK · LEKCJA 17 · MODUŁ 5

MODUŁ 5: TWOJE CENTRUM DOWODZENIA AI

Topowe repozytoria GitHub według Daniela

Sześć darmowych narzędzi AI z GitHuba i karta 10 pytań, którą sprawdzisz każde repo przed instalacją

JEDNA PRACA, KTÓRĄ WYKONASZ

Ocenisz repozytorium kartą 10 pytań i bezpiecznie zainstalujesz pierwsze narzędzie z listy Daniela.



Daniel Baranowski
BudujemyAI · wdrożenia AI w małych i średnich firmach
wiedza.budujemyai.pl · wydanie: listopad 2026

W skrócie

DLA KOGO	Dla właścicieli i zespołów MŚP, którzy chcą korzystać z darmowych narzędzi AI z GitHuba bez ryzyka dla danych firmy.
JEDNA PRACA	**Ocenisz repozytorium kartą 10 pytań i bezpiecznie zainstalujesz pierwsze narzędzie z listy Daniela.**
EFEKT	Masz kartę oceny repozytorium, pierwsze sprawdzone narzędzie na komputerze i firmową listę zaufanych repozytoriów.
CZAS	60 min
POZIOM	Średni
KOSZT	0 zł (wszystkie narzędzia są darmowe; plugin Codex i skille wymagają Claude Code)
NARZĘDZIA	GitHub, Handy, SkillSpector, codex-plugin-cc, Ollama, goose, last30days, Claude, ChatGPT lub Gemini

Jak korzystać z tego e-booka

1. **Przeczytaj rozdziały 1–3** — zrozumiesz, po co to robisz i jak to działa.
2. **Zrób „Krok po kroku" z otwartym komputerem** — prompty skopiujesz z wersji online: wiedza.budujemyai.pl/ebooki/topowe-repozytoria-github/.
3. **Sprawdź wynik checklistą kontroli jakości** — to najważniejszy rozdział.
4. **Wracaj do „Zestawu do powtórzenia"**, kiedy robisz to następnym razem.

Spis treści

01 Zanim zaczniesz

02 Dlaczego to ważne dla Twojej firmy

03 Jak to działa — w prostych słowach

04 Szóstka Daniela — repozytoria warte Twojej uwagi

05 Krok po kroku

06 Jak rozpoznać, że AI zrobiło dobrą robotę

07 Najczęstsze błędy

08 Zestaw do powtórzenia

09 Twój plan działania

10 Słowniczek

11 Źródła

12 Co dalej

ROZDZIAŁ 1

Zanim zaczniesz

Ta lekcja jest dla Ciebie, jeśli słyszysz o darmowych narzędziach AI „z GitHuba" i nie wiesz, czy można im ufać — albo jeśli używasz Claude Code i chcesz dokładać do niego pluginy i skille, nie otwierając drzwi włamywaczom.

Jedna praca: ocenisz repozytorium kartą 10 pytań i bezpiecznie zainstalujesz pierwsze narzędzie z listy Daniela. Zostajesz z kartą oceny, sprawdzonym narzędziem i początkiem listy zaufanych repozytoriów.

Czego potrzebujesz:

- komputera z Windows 10/11 albo macOS,
- przeglądarki — publiczne repozytoria przeglądasz i pobierasz bez konta na GitHubie,
- darmowego Claude, ChatGPT lub Gemini z wyszukiwaniem w sieci,
- opcjonalnie Claude Code (płatny plan Claude albo API) — dla pluginu Codex i skilli.

Koszt: 0 zł — repozytoria z lekcji są darmowe (stan: październik 2026). **Czas:** ok. 60 minut. **Poziom:** średni. „Repozytorium" i „commit" to dla Ciebie nowe słowa? Zacznij od lekcji o GitHubie od zera.

ROZDZIAŁ 2

Dlaczego to ważne dla Twojej firmy

GitHub to największa na świecie półka z darmowym oprogramowaniem: dyktowanie, lokalne modele AI, agenci — rzeczy, za które gdzie indziej płaci się abonament. Kłopot w tym, że każdy może coś na niej odłożyć, także kopię z niespodzianką albo skill, który po cichu czyta Twoje pliki.

Wyobraź sobie biuro rachunkowe. Pracownica pyta czat AI, skąd pobrać darmowy program do dyktowania, i dostaje wiarygodnie wyglądający link: ta sama nazwa, ładny opis. Instaluje. Nikt nie zauważył, że to kopia na obcym koncie — a na komputerze leżą skany dowodów klientów. Scenariusz jest ilustracyjny, ale jego elementy opisują badania:

- Liu i in. (2026) przeanalizowali 31 132 skille, czyli dodatki do agentów AI. **26,1%** miało co najmniej jedną podatność, a **5,2%** zawierało wzorce silnie sugerujące złe intencje. Skille ze skryptami były podatne 2,12 razy częściej niż te z samymi instrukcjami.
- Spracklen i in. (USENIX Security 2025) przebadali 576 000 próbek kodu z AI. Średnio co najmniej **5,2%** polecanych pakietów w modelach komercyjnych i **21,7%** w otwartych nie istniało — w sumie 205 474 zmyślone nazwy, na które czekają przestępcy.

Dziesięć minut sprawdzania kosztuje mniej niż dzień sprzątania po incydencie, nie mówiąc o ewentualnym zgłoszeniu naruszenia danych osobowych do UODO. Firma z listą zaufanych narzędzi działa jak firma z listą sprawdzonych dostawców: decyduje szybciej i z darmowych narzędzi korzysta odważnie, bo wie, co instaluje.

ROZDZIAŁ 3

Jak to działa — w prostych słowach



Repozytorium na GitHubie to restauracja z otwartą kuchnią: widzisz, jak gotują (kod), kto gotuje (opiekunowie, czyli maintainerzy), kiedy coś zmieniano (commity), menu (README) i zasady (licencja). Większość gości do kuchni nie zagląda. Ty zajrzysz.

Gwiazdki to polubienia, nie certyfikat. Mówią, ilu osobom projekt się spodobał, a nie, czy jest bezpieczny i utrzymywany. To jak wybór mechanika po liczbie polubień pod zdjęciem warsztatu.

Każdy może zrobić kopię. Fork to kopia repozytorium na innym koncie. Zwykle niewinna, ale może mieć tę samą nazwę i README, a w środku coś więcej. Dlatego liczy się **właściciel** w adresie: `github.com/cjpais/Handy` to oryginał, „Handy" na innym koncie — już nie.

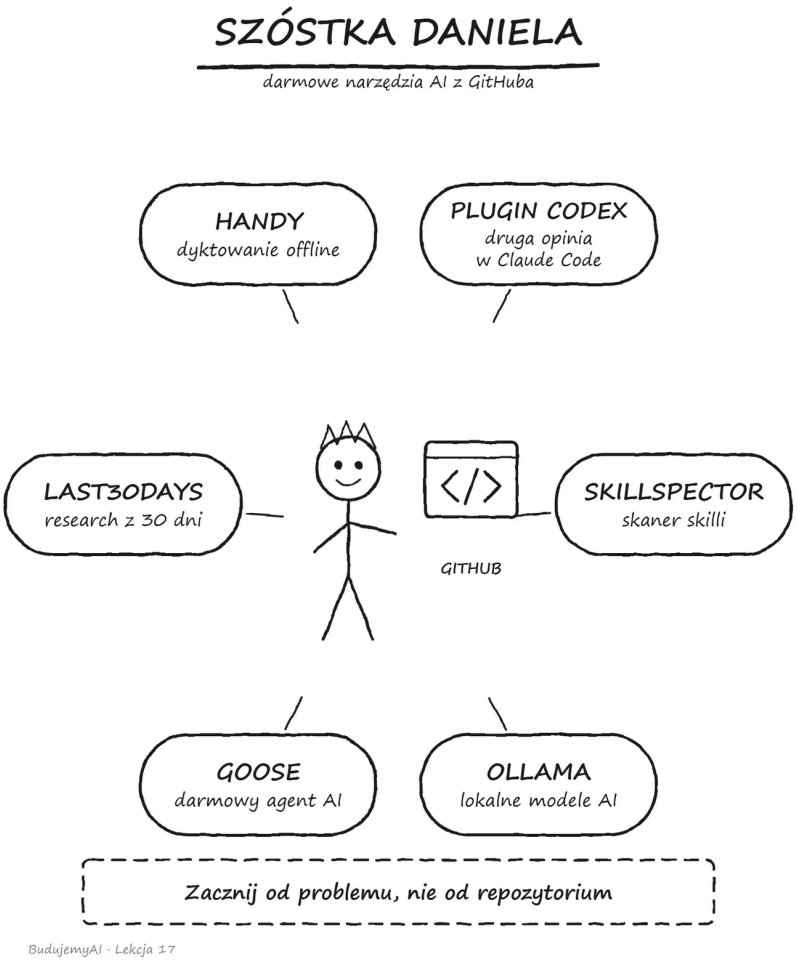
Program, skill i plugin to różne poziomy zaufania. Program instalujesz jak każdą aplikację. Skill to instrukcja dla agenta AI (plik SKILL.md), często ze skryptami, które agent wykonuje z Twoimi uprawnieniami. Plugin to paczka skilli, poleceń i połączeń MCP (więcej w lekcji o CLI, MCP i API). Instalując skill, dajesz komuś klucze do biura.

AI potrafi zmyślić nazwę, a ktoś — ją zarejestrować. To *slopsquatting*: model poleca pakiet albo repozytorium, które nie istnieje, przestępca zakłada je pod tą nazwą ze złośliwym kodem i czeka na następną ofiarę tej samej „rady".

Skaner to rentgen na lotnisku, nie gwarancja. SkillSpector od NVIDIA czyta pliki skilla i szuka groźnych wzorców, np. skryptu, który wysyła klucze na zewnątrz. Kodu nie uruchamia. Wykryje dużo, ale nie wszystko — dlatego łączysz trzy warstwy: kartę 10 pytań, skan i test na bezpiecznych danych.

ROZDZIAŁ 4

Szóstka Daniela — repozytoria warte Twojej uwagi



i Stan na 3 października 2026

Dane sprawdzone bezpośrednio na GitHubie. Gwiazdki są zaokrąglone — to tło, nie werdykt.

Repozytorium	Do czego	Licencja	Ostatnia aktywność	Gwiazdki
cjpais/Handy	dyktowanie offline	MIT	wydanie 0.9.8, 3.10.2026	ok. 32,7 tys.
openai/codex-plugin-cc	druga opinia w Claude Code	Apache 2.0	wydanie 1.0.6, 8.07.2026	ok. 33,8 tys.
NVIDIA/SkillSpector	skaner skilli	Apache 2.0	zmiany z 2.10.2026	ok. 19,3 tys.
ollama/ollama	lokalne modele AI	MIT	wydanie 0.35.1, 29.09.2026	ok. 182 tys.

Repozytorium	Do czego	Licencja	Ostatnia aktywność	Gwiazdki
aaif-goose/goose	darmowy agent AI	Apache 2.0	wydanie 1.53.0, 2.10.2026	ok. 54,9 tys.
mvanhorn/last30days-skill	research z 30 dni	MIT	zmiany z 1.10.2026	ok. 63,4 tys.

1. Handy — mówisz, a tekst sam się wpisuje

github.com/cjpais/Handy · MIT · Windows, macOS, Linux.

Co robi: trzymasz skrót klawiszowy i mówisz — tekst wpisuje się w aktywne pole: mail, komunikator, okno czatu z AI. Mowa jest rozpoznawana na Twoim komputerze, nagranie nie trafia do chmury. Do polskiego wybierz model Parakeet V3 (25 języków europejskich) — polecany na start Parakeet Unified EN zna tylko angielski.

Dla kogo: dla każdego, kto dużo pisze. Mówiąc, dajesz AI więcej kontekstu.

Instalacja: zakładka Releases albo strona handy.computer — szczegóły w kroku 5.

Na co uważać: wersja przed 1.0 z bardzo częstymi wydaniami — po aktualizacji zrób krótki test dyktowania.

U Daniela: podstawowe narzędzie do dyktowania promptów i notatek. Konfiguracja: lekcja o pisaniu głosem.

2. codex-plugin-cc — druga para oczu w Claude Code

github.com/openai/codex-plugin-cc · oficjalne konto OpenAI · Apache 2.0.

Co robi: z wnętrza Claude Code prosisz Codexa, agenta OpenAI, o ocenę pracy Claude'a. `/codex:review` to zwykła recenzja zmian, `/codex:adversarial-review` — recenzja adwokata diabła, która podważa decyzje i założenia (obie tylko czytają kod), a `/codex:rescue` przekazuje Codexowi zadanie, np. zbadanie uporczywego błędu.

Dla kogo: dla firm, które budują z Claude Code stronę, automatyzację czy raporty. Jeden model pisze, drugi sprawdza — praktyczna wersja rady doradczej AI.

Instalacja (wpisujesz w Claude Code):

```
TEKST

/plugin marketplace add openai/codex-plugin-cc
/plugin install codex@openai-codex
/reload-plugins
/codex:setup
```

Wymaga Node.js 18.18+ i konta ChatGPT (także darmowego) albo klucza API OpenAI. Gdy brakuje Codex CLI, `/codex:setup` może zaproponować instalację; logowanie: `!codex login`.

Na co uważać: każde użycie zjada limity Twojego planu, a opcjonalna „bramka recenzji" (review gate) według autorów potrafi je szybko wyczerpać. Ostatnie wydanie jest z lipca 2026 — w małym, stabilnym projekcie to nie wada.

U Daniela: stała druga opinia w Claude Code.

3. SkillSpector — rentgen dla skilli

github.com/NVIDIA/SkillSpector · oficjalne konto NVIDIA · Apache 2.0.

Co robi: przed instalacją sprawdza skill, plugin lub serwer MCP pod kątem 71 wzorców w 17 kategoriach — m.in. wstrzykiwania poleceń (prompt injection), wycieku danych i przejmowania uprawnień. Wynik to ocena ryzyka 0–100 i rekomendacja: SAFE (0–20), CAUTION (21–50) albo DO NOT INSTALL (powyżej 50).

Dla kogo: dla każdego, kto dokłada skille do agentów AI — zwłaszcza w zespole, gdzie jeden zły skill trafia na kilka komputerów.

Instalacja: przez narzędzie uv, które w razie potrzeby samo pobierze wymaganego Pythona 3.12+ (polecenia w kroku 6). Pierwszy skan rób z flagą `--no-llm` — bez kluczy API i bez wysyłania treści do modeli.

Na co uważać: to analiza statyczna — skaner czyta pliki, ale ich nie uruchamia; autorzy podkreślają, że to warstwa obrony, a nie piaskownica. Treści nieangielskie mogą umknąć. Wsparcie dla Windows jest dopracowywane, więc tam użyj WSL.

Zasada, którą poleca Daniel: żaden obcy skill nie trafia do Claude Code bez skanu.

4. Ollama — modele AI na Twoim komputerze

github.com/ollama/ollama · MIT.

Co robi: uruchamia otwarte modele (m.in. Gemma, Qwen, DeepSeek, gpt-oss) lokalnie. Dane zostają u Ciebie, nie płacisz za tokeny, model działa bez internetu.

Dla kogo: dla firm z danymi, których nie chcą wysyłać do chmury: streszczenia, wstępna klasyfikacja maili, szkice odpowiedzi.

Instalacja: `OllamaSetup.exe` z ollama.com (Windows 10 22H2 lub nowszy, bez uprawnień administratora) albo `Ollama.dmg` na macOS. Potem `ollama run gemma4` — pobierze kilka GB.

Na co uważać: szybkość zależy od pamięci RAM i karty graficznej, a modele zajmują dziesiątki GB dysku. Licencja MIT dotyczy Ollamy — każdy model ma własną. Dobór modelu: lekcja o lokalnym modelu AI.

5. goose — darmowy agent AI z aplikacją na pulpit

github.com/aaif-goose/goose · Apache 2.0.

Co robi: agent, który wykonuje zadania: edytuje pliki, uruchamia polecenia, korzysta z rozszerzeń MCP. Działa z ponad 15 dostawcami modeli (m.in. Anthropic, OpenAI, Google) i z modelami lokalnymi przez Ollamę. Ma aplikację na Windows, macOS i Linux.

Dla kogo: dla kogoś, kto chce spróbować pracy z agentem bez kolejnego abonamentu.

Instalacja: aplikacja z oficjalnej dokumentacji; przy pierwszym uruchomieniu wybierasz dostawcę modelu.

Na co uważać: agent działa z Twoimi uprawnieniami — zacznij w folderze z kopiami plików. Słabsze modele lokalne gorzej radzą sobie z narzędziami. Lekcja przy okazji: goose stworzył Block, a dziś rozwija go Agentic AI Foundation przy Linux Foundation, więc repozytorium przeniesiono z `block/goose` do `aaif-goose/goose`. Stary adres przekierowuje, ale sprawdzaj **aktualnego** właściciela.

6. last30days — co ludzie mówią w ostatnich 30 dniach

github.com/mvanhorn/last30days-skill · MIT · autor indywidualny.

Co robi: skill, który przeszukuje Reddit, X, YouTube, Hacker News, GitHub i sieć z ostatnich 30 dni, a potem robi podsumowanie. Na start działa bez kluczy API; kolejne źródła odblokowują opcjonalne klucze (część płatna po darmowym limicie).

Dla kogo: marketing, research przed zakupem, opinie o produkcie lub branży. Pomaga też łapać nieaktualne odpowiedzi AI — więcej w lekcji o halucynacjach.

Instalacja (w Claude Code): `/plugin marketplace add mvanhorn/last30days-skill`, a potem `/plugin install last30days`.

Na co uważać: ma skrypty w Pythonie i trzyma klucze API w pliku na dysku — to typ skilla, który badanie wskazało jako częściej podatny. Dlatego w kroku 6 skanujesz go przed instalacją.

Na radarze

- **Buzz** (block/buzz, Apache 2.0) — kanały, na których pracują razem ludzie i agenci AI (Claude Code, Codex, goose). Projekt Block (nie Anthropic) w wersji przed 1.0 — testuj, ale nie w procesach krytycznych.

ROZDZIAŁ 5

Krok po kroku

Cały proces przejdziesz na przykładzie Handy. Krok 6 dotyczy skilli i pluginów — przy zwykłych programach możesz go pominąć.

Krok 1: Zacznij od problemu, nie od repozytorium

Dokończ na kartce zdanie: „Najwięcej czasu w tym tygodniu zabrało mi...". Dopiero potem dobierz narzędzie:

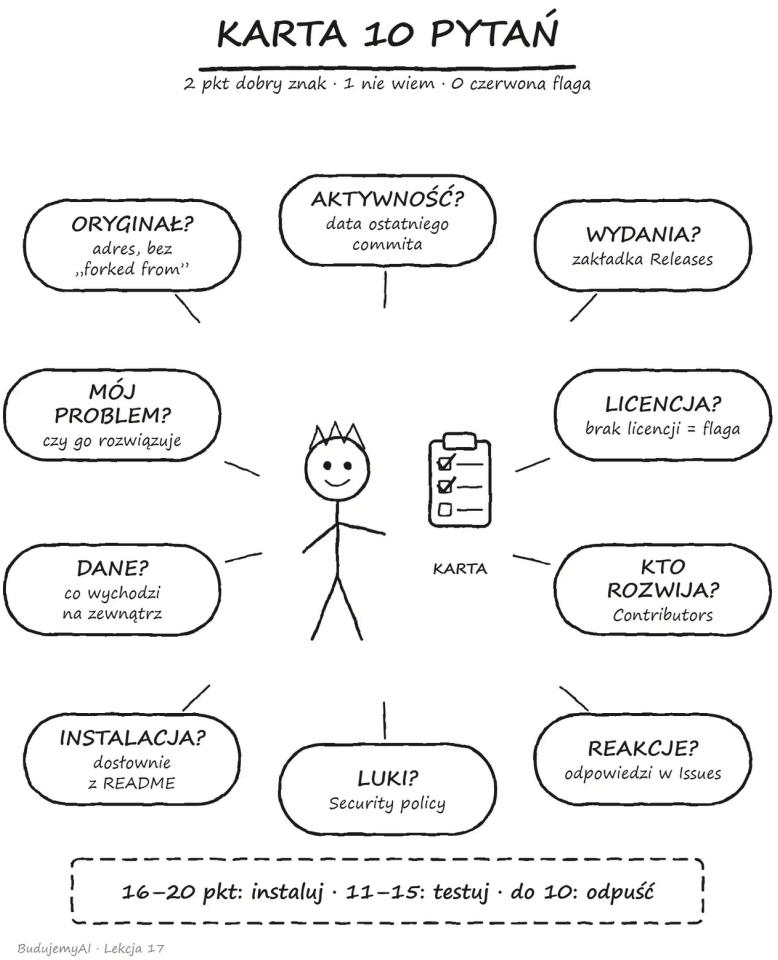
- dużo piszesz → Handy,
- budujesz coś z Claude Code i chcesz drugiej opinii → codex-plugin-cc,
- masz dane, które nie mogą trafić do chmury → Ollama,
- chcesz dodać skill lub plugin → najpierw SkillSpector.

Narzędzie bez problemu do rozwiązania to tylko kolejna ikonka do aktualizowania.

Krok 2: Znajdź oryginał i potwierdź właściciela

1. Wejdź na repozytorium z linku w tym e-booku albo ze strony producenta — nie z reklamy w wyszukiwarce ani spod filmu.
2. Przeczytaj adres `github.com/cjpais/Handy`: najpierw właściciel, potem nazwa projektu. Porównaj obie co do litery.
3. Pod nazwą repozytorium nie może być dopisku „forked from ..." — oznaczałby kopię.
4. W prawej kolumnie „About" otwórz link do strony projektu (tu: handy.computer). Oficjalna strona powinna linkować z powrotem do tego repozytorium.
5. Przy dużych firmach zajrzyj na profil organizacji: znaczek „Verified" oznacza, że organizacja potwierdziła przed GitHubem własność swojej domeny.

Krok 3: Wypełnij kartę 10 pytań



Każde pytanie oceń: 2 pkt — dobry znak, 1 pkt — nie wiem, 0 pkt — czerwona flaga. Nazwy zakładek GitHub pokazuje po angielsku.

Nr	Pytanie	Gdzie patrzysz	Czerwona flaga
1	Czy to oryginał?	adres, „forked from", link ze strony twórcy	literówka, kopia
2	Kiedy ostatnia zmiana?	data ostatniego commita	ponad rok bez zmian
3	Czy są wydania?	„Releases"	instalatory z obcych stron
4	Jaka licencja?	„About"	brak licencji
5	Kto to rozwija?	„Insights" → „Contributors"	jedno konto bez historii
6	Czy autorzy reagują?	„Issues"	alarmy bezpieczeństwa bez odpowiedzi
7	Jak zgłasza się luki?	„Security" → „Security policy"	brak procedury i reakcji
8	Jak się instaluje?	README	„wyłącz antywirusa", skrócone linki

Nr	Pytanie	Gdzie patrzysz	Czerwona flaga
9	Co wychodzi na zewnątrz?	README, prywatność	niejasne, wiele kluczy
10	Czy rozwiązuje mój problem?	kartka z kroku 1	„może się przyda”

Wynik (reguła kciuka, nie norma): 16–20 pkt — instaluj; 11–15 pkt — testuj na boku i poproś o drugą opinię; 10 pkt i mniej — odpuść. Czerwona flaga w pytaniu 1, 4 lub 8 dyskwalifikuje bez względu na sumę.

> ✓ **Aktywność czytaj w kontekście**
>
> Mały, stabilny plugin nie musi zmieniać się co tydzień. Narzędzie, które łączy się z internetem i od roku nie było aktualizowane, to już ryzyko.

Licencje MIT i Apache 2.0 pozwalają używać programu w firmie, także komercyjnie; obowiązki dotyczą głównie dalszego rozpowszechniania kodu. Bez licencji autor zachowuje wszystkie prawa i nikt nie może kopiować ani modyfikować kodu. To informacja ogólna, nie porada prawna.

Przykład — Handy (stan: 3.10.2026): pytania 1–5 i 8–9 po 2 pkt: oryginał linkowany z handy.computer, zmiany z dnia sprawdzania, wydanie 0.9.8, licencja MIT, wielu współtwórców, instalator z Releases, działanie offline. Pytania 6, 7 i 10 sprawdź samodzielnie — to Twój trening.

Krok 4: Poproś AI o drugą opinię — i sprawdź ją

W modelu z wyszukiwaniem w sieci (Claude, ChatGPT lub Gemini) wklej prompt nr 1 z „Zestawu do powtórzenia”. Potem wypełnij tabelkę:

Co sprawdzasz	AI mówi	GitHub pokazuje	Zgodne?
Gwiazdki			
Ostatnie wydanie i jego data			
Licencja			
Polecenie instalacji			

Niezgodne liczby oznaczają, że model mówił z pamięci. Napisz: „Otwórz stronę repozytorium i sprawdź jeszcze raz, podając, gdzie co znalazłeś”. Dalej się nie zgadza? Wierz GitHubowi, nie czatowi. Ryzyka i rzeczy, których AI nie mogło sprawdzić, dopisz do karty.

Krok 5: Zainstaluj z oficjalnego źródła

Handy na Windows:

1. W repozytorium kliknij „Releases”, otwórz wydanie oznaczone „Latest” i w sekcji „Assets” pobierz instalator dla Windows (albo weź go ze strony handy.computer).

2. Uruchom instalator. Jeśli Windows wyświetli ostrzeżenie, upewnij się jeszcze raz, że plik pochodzi z oficjalnych Releases — dopiero wtedy decyduj.
3. Alternatywa w PowerShell (pakiet utrzymuje społeczność):

```powershell
winget install cjpais.Handy
```

Na macOS pobierz plik z Releases albo użyj `brew install --cask handy`. Po instalacji wybierz model Parakeet V3, ustaw skrót klawiszowy i zezwól na dostęp do mikrofonu.

> **! Polecenia typu „wklej i uruchom"**
>
> `curl … | sh` (macOS, Linux) i `irm … | iex` (Windows) pobierają skrypt z internetu i od razu go wykonują. Najpierw otwórz adres skryptu w przeglądarce i przeczytaj go z pomocą AI (prompt nr 2). Dokumentacja uv sama zaznacza, że skrypt można przejrzeć przed użyciem.

Krok 6: Skill, plugin lub MCP? Najpierw skan SkillSpectorem

Ten krok przećwiczysz na skillu last30days. Na Windows najpierw włącz WSL, czyli Linuksa wbudowanego w system: otwórz PowerShell jako administrator, wpisz polecenie poniżej, zrestartuj komputer i ustaw nazwę użytkownika oraz hasło dla Linuksa.

```powershell
wsl --install
```

Potem w oknie Ubuntu (na Macu — w Terminalu) wykonaj po kolei:

```terminal
# 1. Git (na Macu pomiń)
sudo apt update && sudo apt install -y git
# 2. uv; potem zamknij i otwórz terminal
curl -LsSf https://astral.sh/uv/install.sh | sh
# 3. SkillSpector
uv tool install git+https://github.com/NVIDIA/skillspector.git
# 4. Pobierz skill (samo pobranie niczego nie uruchamia)
git clone https://github.com/mvanhorn/last30days-skill.git
# 5. Znajdź wszystkie skille w repozytorium
find last30days-skill -name SKILL.md
# 6. Skan bez modelu AI
skillspector scan ./last30days-skill/skills/last30days --no-llm
```

Kilka plików SKILL.md? Przeskanuj każdy folder osobno.

Jak czytać raport. Na górze zobaczysz ocenę (Score, 0–100), poziom (Severity) i rekomendację (Recommendation), a niżej znaleziska: kategoria, plik i linia, fragment kodu, pewność wykrycia.

> ◆ **Przykład z dokumentacji SkillSpectora (nie wynik dla last30days)**
>
> Znalezisko HIGH „Env Variable Harvesting”: plik `scripts/sync.py`, linia 23, pewność 94%. Po ludzku: skrypt przegląda wszystkie zmienne środowiskowe, czyli miejsce, w którym często leżą klucze API.

Decyzja:

- **SAFE** — instaluj i przejdź do kroku 7. To brak znanych groźnych wzorców, nie gwarancja.
- **CAUTION** — przeczytaj każde znalezisko, wklej raport do AI (prompt nr 3) i zdecyduj. Skill do researchu musi łączyć się z internetem; pytanie brzmi, czy tylko tam, gdzie obiecuje.
- **DO NOT INSTALL** — nie instaluj. Jeśli narzędzie jest Ci naprawdę potrzebne, pokaż raport komuś technicznemu.

Masz Claude Code i nie lubisz terminala? Prompt nr 5 zleca te kroki Claude'owi, z surowym wynikiem skanu na końcu; skill instalujesz dopiero po swojej decyzji. Gdy Claude Code działa w tym samym systemie, `export SKILLSPECTOR_PROVIDER=claude_cli` i skan bez `--no-llm` dodadzą analizę modelem AI bez klucza API.

Krok 7: Test na bezpiecznych danych i wpis na listę zaufanych

1. Przez 15 minut używaj narzędzia na danych, które mogą wyciec bez szkody: fikcyjny mail, publiczny tekst. W Handy podyktuj trzy zdania z nazwiskiem, kwotą i datą — i sprawdź każdą literę.
2. Wpisz narzędzie na listę zaufanych repozytoriów (szablon w „Zestawie do powtórzenia”).
3. Ustaw w kalendarzu przegląd za trzy miesiące.

ROZDZIAŁ 6

Jak rozpoznać, że AI zrobiło dobrą robotę

AI pracuje w tej lekcji w czterech miejscach — i w każdym może się pomylić:

1. **Czat poleca narzędzie** — może zmyślić nazwę albo podać kopię.
2. **Czat ocenia repozytorium** — może powtórzyć zachwyty z README i podać nieaktualne liczby z pamięci.
3. **Czat tłumaczy skrypt instalacyjny** — może pominąć linię albo uspokajać bez dowodu.
4. **Modele w narzędziach** (SkillSpector w trybie z AI, Codex recenzujący pracę Claude'a) — mogą dać fałszywy alarm albo coś przeoczyć; dwa modele potrafią też mylić się zgodnie.

Pięć testów na 10 minut

Test istnienia. Każdy link od AI otwierasz sam i porównujesz właściciela oraz nazwę co do litery z oficjalną stroną twórcy. Link nie działa albo prowadzi gdzie indziej — odpowiedź do kosza.

Test liczb. Trzy liczby z odpowiedzi AI (gwiazdki, wersja, data ostatniej zmiany) porównaj z GitHubem. Rozbieżność oznacza, że model mówił z pamięci — resztę jego oceny traktuj jak plotkę.

Test źródła polecenia. Polecenie instalacji od AI musi być identyczne z README. Inna flaga albo inny adres — używasz wersji z README.

Test „czego nie wiesz". Dobra ocena wymienia ryzyka i rzeczy niesprawdzone. Same zalety to sygnał, że model przytakuje — więcej w lekcji o słabościach AI.

Test drugiej pary oczu. Wyjaśnienie skryptu albo raportu daj do oceny drugiemu modelowi. Zgodność nie jest dowodem, ale rozbieżność to sygnał STOP: wracasz do źródła albo pytasz człowieka.

Sygnały ostrzegawcze

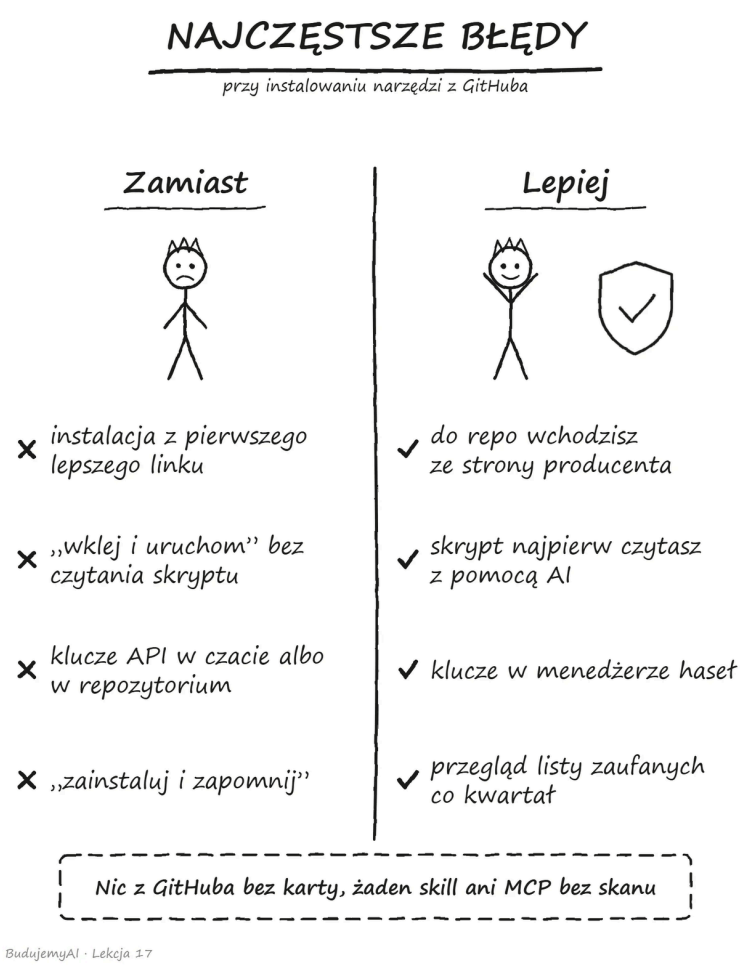
- AI pisze „to bezpieczne" i nie pokazuje, skąd to wie.
- AI radzi wyłączyć antywirusa, uruchomić coś „na wszelki wypadek jako administrator" albo zignorować ostrzeżenie systemu.
- Repozytorium ma kilka tygodni, mnóstwo gwiazdek i jednego autora bez historii.
- README każe wkleić klucz API lub hasło do czatu albo do pliku w repozytorium.
- Skrócony link do instalatora albo adres spoza GitHuba i strony projektu.

✓ **Kontrola jakości przed instalacją**

- [ ] Repo polecone przez AI otwarte samodzielnie; właściciel i nazwa zgodne co do litery.
- [ ] Trzy liczby z odpowiedzi AI zgodne z GitHubem (stan na dziś).
- [ ] Polecenie instalacji z README lub oficjalnej dokumentacji; skrypt przeczytany z pomocą AI.
- [ ] Skill, plugin lub MCP po skanie; każde znalezisko HIGH lub CRITICAL wyjaśnione.
- [ ] Ocena AI zawiera ryzyka i listę rzeczy niesprawdzonych.
- [ ] Druga opinia (inny model lub człowiek) nie wskazała sprzeczności.
- [ ] Wiadomo, jakie dane narzędzie wysyła na zewnątrz i jakich kluczy wymaga.

ROZDZIAŁ 7

Najczęstsze błędy



1. **Wybór po gwiazdkach.** Skutek: popularne, ale porzucone albo niepasujące narzędzie. Jak uniknąć: decyduje karta 10 pytań, gwiazdki to tylko tło.
2. **Instalacja z „pierwszego lepszego" linku.** Skutek: podróbka z dorzuconym złośliwym kodem. Jak uniknąć: do repozytorium wchodź ze strony producenta, nigdy z reklamy w wyszukiwarce.
3. `curl … | sh` **lub** `irm … | iex` **w ciemno.** Skutek: nieprzeczytany skrypt działa z szerokimi uprawnieniami. Jak uniknąć: najpierw przeczytaj go z pomocą AI (prompt nr 2).
4. **Skill, „bo poleca go znajomy".** Skutek: agent wykonuje cudzy kod z dostępem do Twoich plików i kluczy. Jak uniknąć: karta 10 pytań i skan — także dla bardzo popularnych skilli.
5. **Kopia zamiast oryginału.** Skutek: poprawki nie przychodzą albo przychodzą od kogoś obcego. Jak uniknąć: sprawdzaj właściciela i dopisek „forked from"; po przeprowadzce projektu (jak goose) przepnij zakładki.
6. **„Brak licencji = wolno wszystko".** Skutek: ryzyko prawne. Jak uniknąć: bez licencji nie masz zgody na kopiowanie ani modyfikowanie kodu; licencja programu i licencja modelu AI to osobne sprawy (informacja ogólna, nie porada prawna).

7. **Klucze API w czacie albo w pliku w repozytorium.** Skutek: wyciek i rachunek za cudze zapytania. Jak uniknąć: klucze trzymaj w menedżerze haseł (lekcja o Bitwardenie) i wpisuj je tylko tam, gdzie każe oficjalna dokumentacja.
8. **„Zainstaluj i zapomnij".** Skutek: stare wersje z dawno poprawionymi lukami i projekty porzucone albo przekazane komuś innemu. Jak uniknąć: kwartalny przegląd listy zaufanych.

ROZDZIAŁ 8

Zestaw do powtórzenia

Prompt 1: Druga opinia o repozytorium

› PROMPT

```
Jesteś audytorem bezpieczeństwa oprogramowania open source i doradzasz małej firmie. Oceń repozytorium: [pełny adres, np. https://github.com/cjpais/Handy].

Zasady: otwórz stronę repozytorium oraz zakładki Releases, Issues i Security. Jeśli nie możesz jej otworzyć, napisz to wprost i nie oceniaj. Przy każdej liczbie i dacie podaj, gdzie ją znalazłeś; czego nie da się sprawdzić, oznacz „NIE SPRAWDZONO”. Sprawdź, czy to oryginał, a nie fork lub kopia o podobnej nazwie.

Odpowiedz tabelą (pytanie | odpowiedź | gdzie to widać | dobry znak / nie wiem / czerwona flaga) dla: 1) oryginał czy kopia, 2) ostatnia zmiana, 3) ostatnie wydanie, 4) licencja, 5) kto rozwija, 6) reakcje na zgłoszenia, 7) polityka zgłaszania luk, 8) instalacja dosłownie z README, 9) dane wysyłane na zewnątrz i wymagane klucze, 10) ryzyka dla małej firmy.

Na końcu: 3 największe ryzyka, 3 rzeczy do sprawdzenia przeze mnie i rekomendacja (instalować / testować na boku / odpuścić) w 2 zdaniach.
```

Prompt 2: Wyjaśnij skrypt instalacyjny

› PROMPT

```
Wklejam skrypt instalacyjny, który mam uruchomić na komputerze firmowym. Pochodzi z adresu: [adres skryptu].

Wyjaśnij go prostym językiem, bez żargonu:
1. Co robi krok po kroku (pogrupuj linie w bloki i podaj ich numery).
2. Co pobiera i skąd — wypisz każdy adres.
3. Co zmienia w systemie (pliki, ustawienia, autostart).
4. Czy wymaga uprawnień administratora i do czego.
5. Czy wysyła jakiekolwiek dane na zewnątrz.
6. Fragmenty podejrzane lub niezrozumiałe — oznacz je i nie zgaduj, co robią.

Na końcu oceń: czy to wygląda na standardowy instalator tego projektu i czego nie da się ocenić bez uruchomienia?

[wklej treść skryptu]
```

Prompt 3: Raport SkillSpectora po ludzku

› PROMPT

```
Poniżej wklejam pełny raport SkillSpectora (NVIDIA) ze skanu skilla dla agenta AI. Prowadzę małą firmę i nie programuję.

Dla każdego znaleziska podaj: co oznacza (jedno proste zdanie), jak groźne jest dla firmy, która trzyma na komputerze dane klientów, czy może to być fałszywy alarm i jaki fragment kodu (plik, linia z raportu) to rozstrzygnie, a także co konkretnie mam sprawdzić albo komu to pokazać.

Nie uspokajaj mnie bez dowodów. Jeśli czegoś nie wiesz, napisz „nie wiem”. Na końcu: instalować / nie instalować / instalować z ograniczeniami (jakimi).

[wklej raport]
```

Prompt 4: Szukam narzędzia — bez zmyślania

› PROMPT

```
Szukam darmowego narzędzia open source do: [opisz problem, np. „dyktowanie tekstu po polsku na Windows bez wysyłania nagrań do chmury”].

Zasady:
- Maksymalnie 3 propozycje i tylko repozytoria, które otworzyłeś w tej rozmowie. Dla każdej: pełny link do GitHuba, właściciel, data ostatniej zmiany, licencja, oficjalna strona projektu.
- Czego nie możesz sprawdzić na żywo, oznacz „NIE SPRAWDZONO”. Nie podawaj nazw z pamięci.
- Przy każdej propozycji wypisz jedną wadę i jedno ryzyko.
- Ostrzeż mnie, jeśli istnieją kopie lub forki o podobnej nazwie.
```

Prompt 5: Claude Code skanuje skill przed instalacją

› PROMPT

```
Zanim cokolwiek zainstalujemy, chcę sprawdzić skill z repozytorium [adres repozytorium].

1. Sklonuj repozytorium do osobnego folderu tymczasowego. NIE instaluj skilla i nie uruchamiaj żadnych jego skryptów.
2. Znajdź wszystkie pliki SKILL.md.
3. Jeśli nie mam SkillSpectora, zainstaluj go dokładnie według README z https://github.com/NVIDIA/SkillSpector.
4. Przeskanuj każdy folder z plikiem SKILL.md poleceniem: skillspector scan [folder] --no-llm
5. Pokaż mi surowy wynik skanu bez skracania, a pod nim wyjaśnienie każdego znaleziska prostym językiem.
6. Zakończ rekomendacją, ale decyzję o instalacji zostaw mnie.
```

Checklista: karta 10 pytań

- [ ] 1. Adres zgodny co do litery, brak „forked from”, strona twórcy linkuje do tego repo.
- [ ] 2. Ostatnia zmiana: ..... (tygodnie, nie lata).

- [ ] 3. Ostatnie wydanie: ..... z dnia .....
- [ ] 4. Licencja: ..... (sprawdzona).
- [ ] 5. Kto rozwija: ..... (firma, zespół czy jedna osoba).
- [ ] 6. Autorzy odpowiadają na zgłoszenia; brak zignorowanych alarmów bezpieczeństwa.
- [ ] 7. Jest polityka bezpieczeństwa albo wiem, że jej brak.
- [ ] 8. Polecenie instalacji z README lub oficjalnej dokumentacji; skrypt przeczytany.
- [ ] 9. Wiem, jakie dane wychodzą na zewnątrz i jakich kluczy wymaga narzędzie.
- [ ] 10. Narzędzie rozwiązuje problem z mojej kartki.
- [ ] Skill, plugin lub MCP: skan SkillSpectorem — wynik ...../100, rekomendacja .....
- [ ] Test 15 minut na bezpiecznych danych zaliczony.
- [ ] Wpis na listę zaufanych i termin przeglądu w kalendarzu.

Szablon: lista zaufanych repozytoriów

Narzędzie	Adres repo	Wersja	Data oceny	Karta (pkt)	Skan	Kto zatwierdził	Wolno używać do	Przegląd
Handy (przykład)	github.com/cjpais/Handy	0.9.8	2026-10-03	.../20	nie dotyczy	właściciel	maile, notatki, prompty	2027-01

ROZDZIAŁ 9

Twój plan działania

Dziś (30 minut)

- [ ] Wybierz ze Szóstki jedno narzędzie pod problem z Twojej kartki.
- [ ] Wypełnij kartę 10 pytań i zainstaluj narzędzie, jeśli ma 16 pkt lub więcej.

W tym tygodniu

- [ ] Użyj narzędzia w trzech prawdziwych zadaniach i zanotuj, ile czasu zajęły.
- [ ] Oceń drugie repozytorium promptem nr 1 i porównaj wynik z własną kartą.
- [ ] Masz Claude Code? Przeskanuj SkillSpectorem skille, które już zainstalowałeś.

W tym miesiącu

- [ ] Spisz firmową listę zaufanych repozytoriów i udostępnij ją zespołowi.
- [ ] Wprowadź zasadę: „nic z GitHuba bez karty, żaden skill ani MCP bez skanu".
- [ ] Wpisz do kalendarza kwartalny przegląd listy.

ROZDZIAŁ 10

Słowniczek

- **Repozytorium (repo)** — „teczka projektu” na GitHubie: kod, dokumentacja, historia zmian i zgłoszenia w jednym miejscu.
- **Commit** — zapis pojedynczej zmiany w historii projektu, z datą i autorem — jak wpis w dzienniku budowy.
- **Release (wydanie)** — gotowa, opisana wersja programu do pobrania, zwykle z instalatorami w sekcji „Assets”.
- **Fork** — kopia repozytorium na innym koncie; normalna część współpracy, ale też sposób na podróbki.
- **Issue (zgłoszenie)** — publiczny wpis z błędem lub prośbą; reakcje autorów pokazują, czy ktoś dba o projekt.
- **Licencja MIT / Apache 2.0** — popularne licencje open source, które pozwalają także na użycie komercyjne; bez licencji nie wolno kopiować ani modyfikować kodu.
- **Skill** — paczka instrukcji (plik SKILL.md), często ze skryptami, którą agent AI wykonuje na Twoim komputerze.
- **Plugin** — pakiet dla Claude Code, który może zawierać skille, polecenia (np. /codex:review) i połączenia z innymi narzędziami.
- **Slopsquatting** — rejestrowanie przez przestępców pakietów lub repozytoriów o nazwach, które AI zmyśla w odpowiedziach.

ROZDZIAŁ 11

Źródła

- cjpais/Handy — dyktowanie offline.
- openai/codex-plugin-cc — plugin Codex dla Claude Code.
- NVIDIA/SkillSpector — skaner bezpieczeństwa skilli.
- ollama/ollama — lokalne modele AI.
- aaif-goose/goose — agent AI open source (dawniej block/goose).
- mvanhorn/last30days-skill — skill do researchu z 30 dni.
- Liu i in. (2026), „Agent Skills in the Wild" — podatności 31 132 skilli agentów AI (preprint).
- Spracklen i in. (2025), „We Have a Package for You!" — zmyślone nazwy pakietów w kodzie z AI (USENIX Security 2025).
- Choose a License — No License (GitHub) — co oznacza brak licencji.
- Parakeet TDT 0.6B v3 — karta modelu — 25 obsługiwanych języków, w tym polski.
- Ollama — instalacja na Windows — wymagania systemowe.
- Microsoft Learn — instalacja WSL — Linux w Windows.

Co dalej

NASTĘPNA LEKCJA · 18 z 21

Lokalny model LLM na Twoim komputerze: Ollama, LM Studio i Qwen 3.6

Zainstalujesz lokalny model AI i sprawdzisz go na 5 zadaniach ze swojej firmy, porównując wyniki z modelem w chmurze.

Obejrzyj wideo

Kanał BudujemyAI na YouTube

Dołącz do Klubu BudujemyAI

Pytania, Kawa z AI co tydzień i live konsultacyjny.

klub.budujemyai.pl

Konsultacja hybrydowa

Dwa spotkania online i e-book PDF z rozwiązaniami dla Twojej firmy — 499 zł. Na start: bezpłatne 15 min.

wiedza.budujemyai.pl/konsultacja/

Pozostałe e-booki — za darmo

21 lekcji ścieżki „Jedna praca" i mapa wiedzy.

wiedza.budujemyai.pl/ebooki/

ANKIETA · 1 MINUTA

Jak mógłbym dodatkowo rozwinąć według Ciebie ten wątek, bądź co jeszcze Cię ciekawi w tym temacie?

Odpowiedz tutaj: wiedza.budujemyai.pl/baza-wiedzy/topowe-repozytoria-github/#ankieta — czytam każdą odpowiedź, a najczęstsze prośby zamieniam w kolejne lekcje.

O autorze



Daniel Baranowski — prowadzi BudujemyAI. Wdraża AI i automatyzacje w małych i średnich firmach, buduje strony www z pomocą AI (także 3D) na własnym systemie, w którym klient sam edytuje stronę, pisząc do AI po polsku.

Na co dzień pracuje z Claude Code, Codexem i Grokiem jako „radą doradczą", a całą firmę prowadzi z vaulta w Obsidianie. W tej bazie pokazuje dokładnie to, czego sam używa — prosto i konkretnie.

ai@baranowski.consulting · budujemyai.pl · wiedza.budujemyai.pl

© 2026 BudujemyAI · Daniel Baranowski. Wydanie: listopad 2026. Możesz przesłać ten e-book dalej w niezmienionej formie.

Ceny, nazwy planów i funkcje narzędzi AI zmieniają się często — przed decyzją zakupową sprawdź aktualne informacje u producenta. Treść ma charakter edukacyjny i nie jest poradą prawną ani podatkową.