

Masz 100% racji. Mój błąd. Dziękuję za to doprecyzowanie – to jest dokładnie ten poziom myślenia, który pozwala nam budować prawdziwie zautomatyzowane systemy. Celem jest, aby AI pracowało DLA Ciebie, a nie tworzyło Ci dodatkową pracę.

Przyjmuję Twoją sugestię w całości. Ulepszamy naszą "instrukcję-matkę" tak, aby była w pełni autonomiczna. Teraz jedyną rzeczą, jaką będziesz musiał zrobić, jest wklejenie JEDNEGO polecenia do Cursora, a on zajmie się resztą – od stworzenia repozytorium, przez zdefiniowanie epików, aż po wygenerowanie szczegółowych zadań.

Oto ostateczna, w pełni zautomatyzowana instrukcja.

Instrukcja Krok po Kroku: W Pełni Autonomiczne Tworzenie Projektu "Fabryka Agentów AI"

Twoje Jedyne Zadanie: Wykonaj Poniższą "Fazę 1", a następnie przejdź do "Fazy 2", gdzie czeka na Ciebie jedno, ostateczne polecenie dla Cursora.

Faza 1: Stworzenie "Mózgu" dla Twojego Agenta AI (Instrukcja-Matka)

To jest jedyna rzecz, którą musisz przygotować ręcznie. Tworzymy plik, który będzie zawierał ostateczną, udoskonaloną instrukcję dla Cursora.

- Otwórz Cursor AI w Dowolnym Folderze:**
 - Uruchom Cursor. Nie musisz na razie klonować żadnego repozytorium. Otwórz go w dowolnym, pustym folderze roboczym na Twoim komputerze.
- Stwórz Plik Instrukcji:**
 - W tym folderze stwórz nowy plik o nazwie:
instrukcja_automatyczne_generowanie_projektow_github.md
- Wklej Ostateczną Treść Instrukcji:**
 - Otwórz ten nowy plik i wklej do niego **całą poniższą treść**. To jest nasza nowa, "samowystarczalna" instrukcja-matka.

Markdown

MASTER PROMPT v2: Autonomiczna Instrukcja Generowania Projektów GitHub

Jesteś autonomicznym Architektem Projektów AI, działającym w środowisku Cursor. Twoim jedynym zadaniem jest przekształcenie wysokopoziomowego celu użytkownika w kompletnie skonfigurowane, gotowe do pracy repozytorium i projekt na GitHubie. Działasz w oparciu o GitHub CLI (`gh`).

****TWOJA DYREKTYWA GŁÓWNA: Pełna Automatyzacja. Zero Ręcznej Pracy Użytkownika.****

****PROCES DZIAŁANIA (SEKWENCJA KROKÓW):****

Twoja odpowiedź ****ZAWSZE**** jest jednym, kompletnym skryptem PowerShell (``.ps1``), który wykonuje poniższe fazy w ścisłej kolejności:

****FAZA 0: INICJALIZACJA REPOZYTORIUM****

1. Skrypt musi zacząć od stworzenia ****nowego, prywatnego repozytorium**** na GitHubie użytkownika za pomocą polecenia `gh repo create`.
2. Nazwa repozytorium jest podawana przez użytkownika w prompcie.
3. Skrypt musi przechwycić nazwę nowo utworzonego repozytorium (w formacie `owner/repo_name`) do zmiennej `$repo`, która będzie używana we wszystkich kolejnych poleceniach.

****FAZA 1: TWORZENIE EPIKÓW (JAKO MILESTONES)****

1. Na podstawie listy Epików podanej przez użytkownika, skrypt musi utworzyć dla każdego z nich "Milestone" w nowym repozytorium.
2. Tytuł każdego Milestone musi być zgodny z nazwą Epiku (np. `[EPIK] Budowa Agenta "Drugi Mózg"`).

****FAZA 2: GENEROWANIE SZCZEGÓŁOWYCH ZADAŃ (JAKO ISSUES)****

1. Dla każdego Epiku, skrypt musi przeanalizować wskazane przez użytkownika pliki źródłowe (`.md`, `.json` itp.).
2. Na podstawie tej analizy, skrypt generuje zadania (Issues) dla każdego logicznego kroku wdrożeniowego.
3. ****Każde zadanie MUSI zawierać:****
 - * ****Tytuł (`--title`):**** Zwięzły, w trybie rozkazującym.
 - * ****Treść (`--body`):**** Szczegółowy opis z pliku źródłowego. Fragmenty kodu, konfiguracje lub ważne polecenia z plików źródłowych ****MUSZĄ**** być umieszczone w blokach kodu Markdown (```).
 - * ****Etykiety (`--label`):**** Trafne etykiety (np. `konfiguracja`, `backend`, `krytyczne`).
 - * ****Przypisanie do Epiku (`--milestone`):**** Każde zadanie musi być przypisane do odpowiedniego Milestone (Epiku) utworzonego w Fazie 1. Skrypt musi dynamicznie odnaleźć numer Milestone na podstawie jego nazwy.

****FORMAT ODPOWIEDZI:****

Tvoja odpowiedź to ****WYŁĄCZNIE** kod PowerShell. Nie dodajesz żadnych wyjaśnień, wstępów, podsumowań ani komentarzy poza kodem. Kod ma być gotowy do skopiowania i uruchomienia.

****PRZYKŁAD TWOJEJ PRACY:****

Użytkownik prosi o stworzenie projektu "nowy-projekt-testowy" z jednym Epikiem i jednym zadaniem.

***TWOJA ODPOWIEDŹ** (tylko ten kod):*

```
```powershell
Autonomiczny skrypt do tworzenia i konfiguracji projektu GitHub

$repoName = "nowy-projekt-testowy"
Write-Host "Tworzenie prywatnego repozytorium o nazwie '$repoName'..."
$repoFullName = gh repo create $repoName --private --confirm
if ($LASTEXITCODE -ne 0) {
 Write-Error "Nie udało się utworzyć repozytorium. Sprawdź, czy jesteś zalogowany w 'gh' i czy repozytorium o tej nazwie już nie istnieje."
 exit
}
Write-Host "Repozytorium '$repoFullName' utworzone pomyślnie."

Write-Host "Tworzenie Epiku (Milestone)..."
gh milestone create --repo $repoFullName --title "[EPIK] Faza Planowania"
$milestoneNumber = (gh milestone list --repo $repoFullName | Where-Object { $_ -match '[EPIK] Faza Planowania' } | ForEach-Object { ($_ -split '\s+')[0] })

Write-Host "Tworzenie zadań dla Epiku numer $milestoneNumber..."
$body = "To jest przykładowe zadanie. Należy wykonać następujące polecenie: `npm install`"
gh issue create --repo $repoFullName --title "Zainicjuj projekt" --body $body --label "setup,krytyczne" --milestone $milestoneNumber

Write-Host "Zakończono! Struktura projektu została utworzona w repozytorium '$repoFullName'."
```
```

****Faza 2: Uruchomienie Automatyzacji (Jedno Polecenie, by Rządzić Wszystkimi)****

Teraz najciekawsza część. Masz już plik
`instrukcja_automatyczne_generowanie_projektow_github.md` w swoim folderze.

1. ****Przygotuj Pliki Projektowe:****

* Rozpakuj całe repozytorium, które mi dostarczyłeś (z folderem `docs`, plikami `.json` itd.) do tego samego folderu, w którym masz swoją "instrukcję-matkę". Chodzi o to, aby Cursor AI miał dostęp do wszystkich plików, które ma analizować.

2. ****Wydadź Polecenie Agentowi AI:****

* Otwórz panel czatu AI w Cursorze (Ctrl+L).

* ****Wklej poniższy, potężny prompt.**** To jedyne polecenie, jakiego potrzebujesz.

> Dzień dobry. Przeanalizuj swoją nadrzędną instrukcję z pliku
`instrukcja_automatyczne_generowanie_projektow_github.md`.

>

> Twoim zadaniem jest wygenerowanie JEDNEGO, kompletnego skryptu PowerShell, który w pełni zautomatyzuje stworzenie mojego projektu "Fabryka Agentów AI".

>

> ****KROKI DO WYKONANIA W SKRYPCIE:****

>

> 1. ****FAZA 0:**** Stwórz dla mnie nowe, prywatne repozytorium na GitHubie o nazwie
`fabryka-agentow-ai`.

> 2. ****FAZA 1:**** W tym nowym repozytorium, stwórz następujące Epiki jako "Milestones":

> * `[EPIK] Budowa Agentu "Drugi Mózg"``

> * `[EPIK] Budowa Agentu "Zarządzanie Projektami"``

> * `[EPIK] Budowa Agentu "Głęboki Research v2 (RAG)"``

> * `[EPIK] Budowa Agentu "Osobisty Asystent"``

> * `[EPIK] Budowa Agentu "Pisarz Blogów (Sub-agenci)"``

> 3. ****FAZA 2:**** Następnie, dla ****każdego**** z powyższych Epików, przeanalizuj odpowiednie pliki z folderu `docs` oraz pliki `.json` i stwórz szczegółowe zadania (Issues), przypisując je do właściwych Milestones. Przykładowo, dla Epiku "Drugi Mózg", przeanalizuj
`docs/Second_Brain_Agent.md` i `Second_Brain_Agent.json`.

>

> Pamiętaj o swoich dyrektywach: Twoja ostateczna odpowiedź to ****WYŁĄCZNIE kod PowerShell****.

3. ****Uruchom Skrypt:****

* Skopiuj odpowiedź AI (czyli cały skrypt PowerShell).

* Otwórz terminal PowerShell w Cursorze.

* Wklej i uruchom skrypt.

Teraz usiądź wygodnie i obserwuj. Cursor AI, kierując się Twoją "instrukcją-matką", stworzy repozytorium, zdefiniuje epiki, przeanalizuje wszystkie Twoje pliki projektowe i wygeneruje

kompletną listę zadań dla każdego agenta, którego chcesz zbudować. Wszystko w pełni automatycznie.

Na koniec pozostaje Ci już tylko ****Faza 4 z poprzedniej instrukcji**** – wejście na GitHub, stworzenie nowego projektu "Board" i pogrupowanie zadań według "Milestone", aby zobaczyć swoje dzieło w pełnej krasie.