ШІ В ПРОДУКТОВОМУ ДИЗАЙНІ

Claude + міст у Figma. ШІ працює просто у файлах.

Як я налагодив процес, у якому ШІ-агент працює безпосередньо у файлах макетів і дизайн-системи за правилами команди, а останнє слово лишається за дизайнером.

Oleksii Repetii

Менше часу на рутину ✦ ШІ — інструмент, а не заміна ✦ Порядок у дизайн-системі

ПРОБЛЕМА

Рутина забирала в дизайнерів години. Тепер ШІ робить її за хвилини.

01

Сотні компонентів

Перейменування, переприв’язка стилів, Auto Layout і описи — усе вручну, по одному.

02

Перевірки на око

Відступи поза шкалою, чужі стилі й зламані екземпляри знаходять випадково.

03

Знання в голові

Правила ДС ніде не записані, тож новий дизайнер чи розробник їх не бачить.

ІДЕЯ

ШІ робить рутину. Рішення ухвалюють люди.

ДИЗАЙНЕР

Ставить завдання, ухвалює рішення й відповідає за якість. Арт-директор і власник ДС.

ШІ-АГЕНТ

Не просто радить, а працює всередині файлів Figma: робить, перевіряє за правилами й звітує.

ЯК ВЛАШТОВАНО

Шлях завдання: від слів до правки у Figma

1
Дизайнер
Ставить завдання словами — у терміналі чи в чаті проєкту

→

2
ШІ-агент
Claude читає правила й пише невеликий скрипт під завдання

→

3
Локальний міст
Передає скрипт у Figma на комп'ютері дизайнера

→

4
Figma
Скрипт виконується у файлах ДС і макетів

Перед завданням ↓

Агент читає загальні правила й профіль продукту

Після завдання ↑

Звіт дизайнеру: що зроблено, що порушено і що опублікувати

ТЕХНІЧНО

Прямий тунель від ШІ до Figma

АГЕНТ
Claude Code
Пише JS-код на Figma Plugin API під кожне завдання

HTTP →

МІСТ
Локальний сервер
На комп'ютері дизайнера, черга на кожен файл

WebSocket →

FIGMA
Плагін-виконавець
Відкритий у Figma Desktop і виконує код у файлі

Звідки приходять завдання

З термінала на Mac — напряму. З хмарного проєкту — через сесію віддаленого керування на тому ж Mac.

Що повертається

Результат, лог і помилки з підказками. Агент перевіряє їх і складає звіт.

Готові скрипти

Типові дії не пишуться щоразу з нуля: перевірка макета за правилами, список компонентів ДС, стани завантаження.

ЧОМУ ТАК

Без лімітів стандартного Figma MCP

	СТАНДАРТНИЙ MCP	ПРЯМИЙ МІСТ
Ліміти	200 викликів на день; 6 на місяць для місць View і Collab	✓ Немає — усе працює локально
Запис у файл	Здебільшого читає макет; запис — лише в бета-версії	✓ Створює, змінює, прив’язує стилі
Масштаб	Окремі виклики, один за одним	✓ Сотні шарів одним скриптом
Кілька файлів	Один контекст	✓ Паралельно, з чергою на кожен файл

Ліміти Figma — за документацією станом на жовтень 2026 р. Обмеження моста: має бути відкрита Figma Desktop, бібліотека публікується вручну.

НЕЗАЛЕЖНІСТЬ ВІД МОДЕЛІ

Модель можна змінити. Система лишається.

ЗМІННА ЧАСТИНА

Модель ШІ

- **Claude — зараз**
- **Інша модель — завтра**
- **Нова версія — щойно вийде**

Підійде будь-яка модель, що вміє писати код і працювати з інструментами

ПОСТІЙНА ЧАСТИНА

Конструкція процесу

- Правила й профілі продуктів
- Гейти й аудит
- Міст у Figma
- Готові скрипти
- Пам’ять і журнал рішень
- Рішення дизайнера

Не змінюється: процес, правила, перевірки й файли

Залежить від моделі: якість рішень і швидкість

КЛЮЧОВА ІДЕЯ

Розум процесу — у правилах, а не в інструменті

Загальні правила	Для будь-якого продукту: як працюємо, що можна й чого не можна, реєстр продуктів
Профіль продукту	Сітки, шкали відступів і радіусів, шрифти, правила збирання екранів, перевірки, винятки
Пам’ять проєкту	Рішення між завданнями: «це вже вирішили, більше не пропонувати»

ГЕЙТИ ПЕРЕВІРКИ

0 **План із ДС**
Компоненти й стилі — до першої команди

1 **Правила збирання**
Лише екземпляри, стилі й шкали ДС

2 **Аудит**
Скрипт перевіряє зібране й дає звіт

3 **Винятки**
Усе, чого немає в ДС, — кандидат на рішення

ЯК ЦЕ ПРАЦЮЄ

Два потоки завдань

ПРАВКИ В ДС

1 Завдання
Словами, з контекстом і метою

2 Правила
ШІ читає правила й профіль

3 Пілот
Бекап і проба на одному наборі

4 «Так»
Дизайнер переглядає й вирішує

5 Масово
Правка всього набору й аудит

6 Звіт
Підсумок і список «опублікувати»

НОВІ ЕКРАНИ

1 План із ДС
Що беремо з ДС і чого в ній бракує

2 Збирання
За правилами й шкалами ДС

3 Аудит
Скрипт перевіряє зібране

4 Варіанти
Кожен поруч, в окремому фреймі

5 Рішення
Дизайнер обирає варіант і що з нового додати в ДС

Рішення одразу записується в профіль продукту, тож наступного разу ШІ його вже знає.

ДВА ВХОДИ

Дрібне — у термінал, велике — у проєкт

Термінал на комп'ютері

- Точкова правка одного компонента
- Питання про виділене у Figma
- Швидка перевірка
- Дешевше й швидше

Хмарний проєкт

- Чистка розділів ДС за планом
- Аудити зі звітом, збирання екранів
- Окрема гілка й свіжий контекст на кожне завдання
- Завдання можна ставити з телефона

Обидва входи читають ті самі правила — тому результат однаковий.

РОБОЧИЙ ДЕНЬ

Увімкнув — попрацював — вимкнув

09:00

Старт

Одна команда запускає міст, і плагін відкривається у Figma

→

увесь день

Робота з ШІ

Дрібні правки — у терміналі. Великі завдання — у хмарному проєкті, кожне у своїй гілці. Завдання можна поставити з телефона.

→

після

Фіксація

Публікація бібліотеки за списком, нові правила — у git

→

18:00

Стоп

Міст вимикається однією командою

Правила, профілі та скрипти лежать у приватному git-репозиторії: видно історію, можна відкотитися.

МУЛЬТИПРОЄКТНІСТЬ

Новий продукт — це новий профіль, а не новий процес

1

Знайомство

Продукт, файли, флоу

2

Опис ДС

ШІ читає файл, нічого в ньому не змінюючи

3

Профіль

Із шаблону: шкали, сітки, правила

4

ДС під ШІ

Описи, єдині назви й стани, щоб ШІ розумів систему

5

Генерація

Індекс, шаблони, перевірки

Один набір загальних правил і окремий профіль для кожного продукту. Продукти з різними правилами працюють паралельно й не змішуються.

ГЕНЕРАЦІЯ ЕКРАНІВ

Дизайн-система — основа, а не рамка

Кандидат у ДС — новий елемент, якого ще немає в дизайн-системі. ШІ робить його на стилях ДС, пояснює, навіщо він потрібен, і чекає рішення дизайнера.

1
Шукаємо рішення
Найкраще для екрана, а не лише з наявного

→

2
Бракує деталі
ШІ збирає її на кольорах, шрифтах і відступах ДС

→

3
Позначка
«Кандидат у ДС» і пояснення, навіщо він

→

4
Рішення
Дизайнер вирішує: брати чи ні

→

5
У систему
Схвалений кандидат стає компонентом ДС

Приклад: екрану потрібен таймер, а в ДС його немає. ШІ робить таймер у стилі системи, а ви вирішуєте, чи стане він компонентом.

Кольори, шрифти, відступи й сітки — лише з ДС. ШІ має свободу, а система росте керовано.

РОЗВИТОК

Кожна складність стає правилом

1

Складність

Щось пішло не так або вийшло задорого

→

2

Розбір

Шукаємо причину, а не винного

→

3

Правило

Формулюємо так, щоб ШІ міг його виконати

→

4

Запис

У загальні правила або профіль продукту

📖 Усі випадки ведемо в журналі «проблема → як розв’язали → правило». З кожним завданням процес стає надійнішим.

ПРИНЦИПИ

Правила, які захищають файли

Бекап перед правкою

Завжди є куди відкотитися

Спершу проба на малому

Один набір, потім — масово

Без «так» нічого не змінюється

Останнє слово — за дизайнером

Не видаляємо, поки не перенесли

Макети не ламаються

Рішення одразу записуємо

ШІ не забуває домовленостей

Одне завдання — одна сесія

ШІ не робить зайвого

ПІДСУМОК

Вирішує дизайнер. Робить ШІ. Правила — спільні.

Процес записаний у правилах і профілях, тому його можна повторити на будь-якому продукті.

Telegram @gnt_alex

repetii.space

Oleksii (Alex) Repetii · Product Design Lead