

Ekosystemy Human-AI w modelu „wioski kosmicznej” – analiza pięciu habitatów

Wprowadzenie

Rozwój złożonych **ekosystemów human-AI** inspirowanych koncepcją „wioski kosmicznej” staje się coraz bardziej realny w nauce i biznesie. Pojęcie „wioski kosmicznej” odnosi się do zintegrowanych habitatów – czy to w przestrzeni kosmicznej, czy na Ziemi – gdzie ludzie i zaawansowane systemy AI współistnieją w zamkniętym, samowystarczalnym środowisku, współpracując dla osiągnięcia wspólnych celów. Przykładem jest projekt **EBIOS** (Experimental BIOregenerative Station), czyli **turystyczna „wioska kosmiczna”** symulująca warunki poza Ziemią – autonomiczna baza dla 100 osób, zdolna do samodzielnej produkcji energii, wody i żywności ¹. Tego typu habitaty traktować można jako **źródła danych dla AI** i jednocześnie jako **struktury społeczno-ekonomiczne** służące przetwarzaniu ryzyka, informacji i kompetencji.

Niniejszy raport przedstawia **pięć zidentyfikowanych modeli ekosystemów Human-AI** osadzonych w realiach „wioski kosmicznej” lub jej ziemskich analogów. Każdy model omówiono pod kątem założeń, mechanizmów działania oraz aspektów ekonomicznych i organizacyjnych. Analizie poddano następujące modele habitatów Human-AI:

1. **Laboratorium Społecznej SI (Social-AI Lab)** – otwarte środowisko współpracy społeczności i AI.
2. **AI-ops jako usługa (AI-ops-as-a-service)** – habitat świadczący usługi operacji AI na żądanie.
3. **Prototypownia danych (Data Prototyping Lab)** – środowisko szybkiego prototypowania rozwiązań opartych o dane.
4. **Habitat regeneracyjny** – samowystarczalny ekosystem z AI w pętli, analog zamkniętych systemów podtrzymania życia.
5. **Środowisko certyfikacji i testów (Benchmarking & Certification Hub)** – habitat dedykowany walidacji, benchmarkom i certyfikacji systemów AI.

Dla każdego ekosystemu dokonano wielowymiarowej charakterystyki obejmującej: **(1)** definicję i założenia modelu, **(2)** źródła wartości ekonomicznej, **(3)** rachunek opłacalności (CAPEX/OPEX, ryzyka, progi rentowności), **(4)** mechanizmy redukcji kosztów i maksymalizacji wartości (np. pętle danych, autonomia, cyfrowe feromony), **(5)** rolę człowieka w pętli (HITL), **(6)** przepływ danych od surowych do produktów premium oraz **(7)** krytyczne heurystyki sukcesu (utrzymanie wartości danych, unikanie pułapek strukturalnych).

W toku analizy przywołano **analogiczne systemy rzeczywiste**, m.in. **ISS National Lab** (Międzynarodowa Stacja Kosmiczna jako laboratorium), europejski projekt **MELISSA** (zamknięty system podtrzymania życia), program **NASA Flight Opportunities** (środowisko testów w warunkach kosmicznych), modele biznesowe **OpenAI** (licencjonowanie AI), rozwiązania geoinformatyczne **Maxar** (dane satelitarne jako usługa) czy mechanizmy finansowania innowacji **NASA SBIR/STTR**. Zastosowano także odwołania do uznanych metodyk, takich jak **Technology Readiness Levels (TRL)** dla oceny dojrzałości technologii, miary efektywności ekonomicznej jak **ROI** (zwrot z inwestycji) i **NPV** (netto obecna wartość projektu), zasady zarządzania danymi **FAIR** oraz mechanizmy samoorganizacji jak **stigmergia**.

Raport ma charakter naukowy – prezentuje **usystematyzowane porównanie** modeli (również w formie tabelarycznej) i poparty jest aktualnymi danymi liczbowymi oraz literaturą przedmiotu. Wszystkie źródła informacji zostały oznaczone w tekście przypisami w formacie **[źródłolinia]** , zgodnie z zasadami rzetelności naukowej.

Ekosystem 1: Laboratorium Społecznej SI (Social-AI Lab)

Definicja i założenia modelu

Laboratorium Społecznej SI to habitat, który łączy lokalną społeczność z technologiami sztucznej inteligencji w celu wspólnego eksperymentowania i rozwoju odpowiedzialnych rozwiązań. Model ten zakłada otwarte, **partycypacyjne środowisko badań i wdrożeń AI**, gdzie obywatele, eksperci i algorytmy uczą się od siebie nawzajem. Przykładem jest niedawno otwarte **Social AI Lab w Rotterdamie** – kreatywna przestrzeń, w której mieszkańcy, naukowcy, projektanci, studenci i decydenci **wspólnie eksplorują zastosowania AI**, zwłaszcza w obszarach społecznych takich jak opieka, edukacja czy usługi miejskie ² . Kluczowym założeniem jest *inkluzyjność* i *odpowiedzialność*: technologia ma być rozwijana **z udziałem ludzi, których dotyczy**, a nie tylko dla nich, aby lepiej odpowiadała na realne potrzeby i wartości społeczne ³ . Laboratorium kieruje się zbiorami zasad etycznych (godność, prywatność, transparentność, bezpieczeństwo), które wyznaczają ramy projektów i eksperymentów ⁴ .

W praktyce habitat tego typu przyjmuje formę „**living labu**” – fizycznej przestrzeni (np. lokalnego centrum innowacji czy pracowni na uczelni) sprzężonej z wirtualnymi platformami, gdzie gromadzone są dane społeczne i testowane prototypy AI. Założenie jest takie, że *człowiek jest stale w pętli rozwoju AI*, od definiowania problemów po ewaluację rozwiązań. Model zakłada ponadto otwartość na nowych uczestników (np. poprzez warsztaty, hackathony, mobilne lab-busy odwiedzające dzielnice ⁵) oraz **wymianę wiedzy** w duchu open source. Dzięki temu laboratorium pełni jednocześnie funkcję edukacyjną i integracyjną – zwiększa **kompetencje cyfrowe społeczności** i buduje zaufanie do AI.

Źródła wartości ekonomicznej

Choć na pierwszy rzut oka laboratorium społeczno-SI nie generuje tradycyjnego produktu, ma ono liczne **źródła wartości informacyjnej i ekonomicznej**:

- **Dane i insighty społeczne:** W toku warsztatów i pilotowych wdrożeń zbierane są unikalne dane o potrzebach mieszkańców, ich zachowaniach i preferencjach względem technologii. Te *dane behawioralne i kontekstowe* są cennym zasobem – pozwalają lepiej trenować lokalne modele AI (np. asystentów miejskich) i stanowią bazę wiedzy dla polityk publicznych. Odpowiednio anonimizowane i agregowane, mogą posłużyć do tworzenia raportów lub usług konsultingowych dla innych miast.
- **Kompetencje i edukacja:** Laboratorium wytwarza **kapitał ludzki** – szkoli mieszkańców i urzędników w zakresie AI. Długofalowo przekłada się to na wzrost innowacyjności regionu i **wyższą produktywność** (np. lokalne startupy mogą korzystać z wyszkolonych kadr). Wartość ekonomiczna jest tu pośrednia: lepiej wyedukowane społeczeństwo szybciej adoptuje nowe technologie, co zmniejsza koszty transformacji cyfrowej w przyszłości.
- **Prototypy i usprawnienia usług:** Projekty powstające w Social-AI Lab to często prototypy rozwiązań usprawniających usługi publiczne (np. chatboty do informacji miejskiej, modele do alokacji zasobów w opiece społecznej). Jeśli któryś prototyp okaże się udany, może zostać wdrożony na szerszą skalę, generując **oszczędności** (bardziej efektywne świadczenie usług) lub **nowe przychody** (np. udostępnienie narzędzia innym gminom na zasadzie licencji).
- **Benchmarking społeczny:** Poprzez testowanie AI na realnych użytkownikach, laboratorium dostarcza informacji o tym, *jak algorytmy radzą sobie w prawdziwym środowisku społecznym*. Takie **benchmarki użyteczności i zaufania** mają wartość dla twórców technologii – firmy mogą np. finansować testy swoich rozwiązań w labie, aby uzyskać certyfikat „przetestowane z udziałem społeczności”.

Ponieważ laboratorium działa zwykle non-profit lub w modelu partnerstwa publiczno-prywatnego, **przychody** mogą pochodzić z grantów (np. programy badawcze UE, fundusze miejskie na innowacje), ze **sponsorowanych projektów R&D** (firmy płacą za możliwość przetestowania prototypu w kontrolowanych warunkach społecznych) oraz z **produktów wiedzy** (raporty z badań, szkolenia, licencjonowanie udanych aplikacji). Przykładowo, gmina Rotterdam finansuje Sociaal AI Lab jako element programu integracji cyfrowej mieszkańców ⁶, co w założeniu ma przynieść wartość w postaci zmniejszania wykluczenia cyfrowego i kosztów z nim związanych.

Rachunek opłacalności (CAPEX, OPEX, ryzyka)

Nakłady inwestycyjne (CAPEX) dla tego modelu są relatywnie niewielkie w porównaniu z infrastrukturą kosmiczną – obejmują wyposażenie przestrzeni (komputery, urządzenia IoT, serwery do lokalnego przetwarzania danych), ewentualnie adaptację budynku na potrzeby labu oraz koszty stworzenia platformy danych. Często wykorzystuje się istniejącą infrastrukturę (np. pomieszczenia szkolne, biblioteki), by ograniczyć CAPEX. Przykładowo, rotterdamskie laboratorium zostało ulokowane w centrum sąsiedzkim Hilleliet i korzysta z zasobów partnerów akademickich ⁷, co obniża koszty początkowe.

Koszty operacyjne (OPEX) to głównie wynagrodzenia interdyscyplinarnego zespołu (trenerzy, analitycy danych, moderatorzy warsztatów, inżynierowie AI), utrzymanie sprzętu i oprogramowania oraz organizacja ciągłych działań (warsztatów, hackathonów, kampanii informacyjnych). Część tych kosztów można minimalizować poprzez wolontariat lub współpracę z uczelniami – np. studenci wykonują prace w ramach praktyk, a naukowcy traktują udział jako element swoich badań. Ryzykiem OPEX są jednak trudno mierzalne rezultaty: **uzasadnienie wydatków** przed decydentami wymaga wykazania konkretnych efektów (np. liczba przeszkolonych osób, liczba prototypów wdrożonych przez miasto).

Ryzyka finansowe i operacyjne obejmują:

- *Ryzyko braku ciągłości finansowania*: Jeśli projekty nie przyniosą szybko widocznych korzyści, sponsorzy (np. władze miasta) mogą ograniczyć finansowanie. Model opiera się na długofalowych korzyściach społecznych, co może być politycznie wrażliwe.
- *Ryzyko technologiczne*: Tworzone prototypy mogą się nie skalować poza warunki labowe, przez co inwestycja w nie nie zwróci się.
- *Ryzyko społeczne*: Utrata zaufania społeczności (np. przez incydent naruszenia prywatności danych uczestników) mogłaby zniweczyć misję labu.

Punkt równowagi w tym modelu jest osiągnięty raczej w kategoriach *społecznego ROI* niż bezpośredniego zysku finansowego. Analiza kosztów i korzyści uwzględnia zatem wskaźniki takie jak **Social Return on Investment (SROI)** – np. wartość zaoszczędzona na usługach społecznych dzięki lepszymu dopasowaniu ich do potrzeb mieszkańców, czy **wzrost jakości życia** trudny do wyrażenia w PLN. W dłuższej perspektywie laboratorium może jednak wpływać na budżet pozytywnie: zmniejszając ryzyko nietrafionych inwestycji w e-administrację (bo rozwiązania są najpierw prototypowane z użytkownikami), *unikamy kosztów „pętli długu technologicznego”* polegającej na poprawianiu nieudanych systemów. Jak wskazują badania nad programami innowacji społecznych, inwestycje takie mogą mieć mnożnik ekonomiczny – np. programy typu SBIR/STTR generują w gospodarkach ROI rzędu **14,7:1** (każdy \$1 wydany na badania przynosi \$14,7 wartości ekonomicznej) ⁸, choć tu znaczną część zysku stanowią trudno uchwytne efekty pośrednie.

Mechanizmy redukcji kosztów i maksymalizacji wartości

Laboratorium Społecznej SI wykorzystuje szereg mechanizmów, by **działać efektywnie ekonomicznie i wartościowo**:

- **Ekologia innowacji (symbioza z otoczeniem):** Lab funkcjonuje w ekosystemie lokalnym – współpracuje ze szkołami, NGO, startupami. Tworzy to *symbiotyczną relację*: z jednej strony laboratorium czerpie od partnerów zasoby (np. dane od instytucji miejskich, wolontariuszy z uczelni), z drugiej – oddaje im wypracowane rozwiązania i kompetencje. Taka *cyrkulacja zasobów* ogranicza marnotrawstwo – np. dane zebrane w jednym projekcie mogą być ponownie użyte (zgodnie z zasadami **FAIR – Findable, Accessible, Interoperable, Reusable**) w kolejnym kontekście ⁹.
- **Autonomia i automatyzacja:** Choć istotą labu jest udział ludzi, stosuje się narzędzia automatyczne do zmniejszania powtarzalnych kosztów. Przykładowo, część analiz danych odbywa się z użyciem gotowych **platform AutoML**, a monitoring nastrojów społeczności (np. analiza opinii) może być zautomatyzowany. Pozwala to małemu zespołowi obsłużyć większą liczbę projektów.
- **„Pętle danych” i uczenie się na feedbacku:** Projekty prowadzone są iteracyjnie – pierwsza faza pilota dostarcza danych zwrotnych (np. co nie działa w prototypowej aplikacji), te dane są analizowane i prototyp jest poprawiany. Ta szybka *pętla uczenia się* minimalizuje koszt porażki: błędy wyłapywane są wcześniej, na małej próbie, zanim zostaną popełnione na szeroką skalę. Ogranicza to ryzyko wydania środków na pełne wdrożenie nietrafionego rozwiązania. Mechanizm ten przypomina **metodę agile** w IT – krótkie sprinty rozwojowe z ciągłą walidacją.
- **Cyfrowe feromony i stigmergia:** Ciekawą metaforą organizacyjną są *cyfrowe feromony* – laboratorium może wykorzystywać systemy oznaczania problemów lub pomysłów, które wymagają uwagi społeczności. Przykładowo, gdy uczestnicy warsztatów identyfikują ważny lokalnie problem (np. bezpieczeństwo pieszych), pozostawiają w systemie „znacznik” – sygnał, który przyciąga zainteresowanie innych (na zasadzie feromonów zostawianych przez mrówki). W algorytmicznych analogiach **wirtualne mrówki eksplorują przestrzeń problemu, zostawiając cyfrowe feromony na ścieżkach – częściej uczęszczane ścieżki z większą ilością feromonu uznawane są za lepsze rozwiązania** ¹⁰. Społeczność skupia się zatem na tematach o najsilniejszych sygnałach oddolnych, co zwiększa trafność i wartość projektów, a redukuje zaangażowanie w inicjatywy o niskim poparciu (unikamy marnowania środków na „bezużyteczne dane” i pomysły bez społecznej akceptacji).
- **Kompetencje AI na rzecz redukcji kosztów:** Paradoksalnie, samo AI jest wykorzystywane w labie do usprawniania pracy labu. Przykładowo, **chatboty wspierają moderatorów** odpowiadając na często zadawane pytania uczestników warsztatów, systemy rozpoznawania mowy automatycznie transkrybują i tagują dyskusje, a narzędzia do analizy sieci społecznych pomagają wyłonić kluczowych liderów opinii do włączenia w projekty. To redukuje koszty osobowe i zwiększa skalowalność działań bez utraty jakości.

Wszystkie te mechanizmy sprawiają, że **Laboratorium Społecznej SI jest w stanie maksymalizować wartość społeczną z każdej zainwestowanej złotówki**, jednocześnie minimalizując koszty nieudanych eksperymentów poprzez wczesne i stałe korygowanie kursu. Innymi słowy, lab stosuje filozofię „**działaj mądrze, działaj oszczędnie**” – najpierw wykorzystaj to, co już masz (dane, wiedzę ludzi), i reaguj adaptacyjnie na informacje zwrotne.

Rola człowieka w pętli (HITL)

W Laboratorium Społecznej SI **człowiek jest centralnym ogniwem pętli uczenia i działania** – od początku do końca procesu:

- **W fazie definiowania problemu** to ludzie (mieszkańcy, eksperci dziedzinowi) określają, jakie wyzwania są istotne. AI może wspomagać analizę danych o problemie, ale to człowiek nadaje kierunek i priorytety zgodnie z wartościami społecznymi.
- **W fazie rozwoju rozwiązania** ludzie uczestniczą jako współtwórcy – np. mieszkańcy dostarczają *treningowych danych* (choćby poprzez oznaczanie zgłoszeń w aplikacji, co jest formą ręcznego labelingu), a eksperci nadzorują poprawność działania modeli (HITL jako walidator jakości). Gdy AI model wygeneruje propozycję decyzji czy rekomendację, człowiek może ją ocenić i skorygować. W ten sposób następuje **szkolenie z rewizją przez człowieka** – analogicznie jak w podejściu *reinforcement learning from human feedback*, gdzie preferencje ludzi służą za sygnał nagrody do dostrajania modelu ¹¹ ¹².

- **W fazie wdrożenia i użytkowania** człowiek pozostaje „strażnikiem” wartości – monitoruje działanie AI w realnym świecie, wychwytyjąc potencjalne błędy lub skutki uboczne. Na przykład opiekunowie merytoryczni sprawdzają, czy chatbot miejski udziela zgodnych z prawdą i pomocnych odpowiedzi. Gdy AI popełni błąd (np. udzieli nieodpowiedniej porady seniorowi), interwencja człowieka koryguje sytuację i zarazem wskazuje systemowi do nauki, że taka odpowiedź była niepożądana.

Człowiek w pętli **zwiększa wartość** rozwiązania przede wszystkim poprzez wniesienie *kontekstu i etyki*, których AI może nie posiadać. Minimalizuje też ryzyko błędów strukturalnych – np. zapobiega przejęciu stronniczości danych treningowych, dzięki świadomemu nadzorowi. Ponadto, ludzie pełnią rolę **inicjatorów rozwoju**: to często oddolne pomysły mieszkańców („a może wykorzystamy AI do optymalizacji zużycia energii w naszej dzielnicy?”) stają się zaczątkiem nowych projektów. Laboratorium stwarza platformę, gdzie te inicjatywy mogą zostać podchwyczone i rozwinięte z pomocą narzędzi AI.

Jest to więc model *Human-in-the-Loop* w pełnym znaczeniu – nie tylko jako adnotator danych czy awaryjny operator, ale jako **współpartner AI**. Przykłady z Rotterdamu pokazują, że taka współpraca może prowadzić do bardziej efektywnych rozwiązań: np. cyfrowy asystent miejski „Digiderius” powstał w laboratorium poprzez rozmowy mieszkańców z prototypem, co pozwoliło go dostosować do lokalnych realiów¹³. Tam, gdzie sama automatyka by zawiodła lub była nieakceptowalna społecznie, człowiek w pętli zapewnia *legitymizację i adaptację* technologii.

Przepływ danych i produkty informacyjne

W ekosystemie laboratorium społecznego przepływ danych można przedstawić jako **wielostopniowy łańcuch wartości danych**:

1. **Dane surowe**: Na wejściu znajdują się dane pierwotne pozyskiwane od społeczności i środowiska lokalnego. Mogą to być wyniki ankiet, wpisy z platform zgłaszania problemów miejskich, dane sensorów miejskich (hałas, zanieczyszczenia) czy choćby notatki z wywiadów fokusowych. Dane te są często nieustrukturyzowane i heterogeniczne.

2. **Metadane i wstępna obróbka**: Surowe dane są oczyszczane i wzbogacane metadanymi – np. tagami kategorii problemu, znacznikiem czasu i miejsca. Tworzy się np. **bazy wiedzy** o problemach: każdy wpis obywatela (np. „latarnia nie działa na ulicy X”) dostaje kategorie (bezpieczeństwo, infrastruktura) i priorytet nadany przez społeczność (poprzez głosowania). Te metadane pozwalają AI lepiej zrozumieć kontekst i ułatwiają wyszukiwanie podobnych przypadków.

3. **Weryfikacja i walidacja**: Zanim dane posłużą do trenowania modeli czy podejmowania decyzji, są walidowane – tu ponownie istotna jest rola ludzka. Moderatorzy lub zautomatyzowane skrypty sprawdzają np., czy zgłoszenie nie jest powieleniem, czy nie zawiera wulgaryzmów, czy sensor nie dał odczytu błędnego (kalibracja). Walidacja zapewnia **jakość danych** – dane błędne lub nieprzydatne są odsiewane, co zapobiega karmieniu AI śmieciowymi informacjami (zasada „garbage in, garbage out”).

4. **Trenowanie modeli / generowanie insightów**: Na zweryfikowanych danych uruchamiane są algorytmy. W zależności od celu, może to być trenowanie lokalnego modelu ML (np. klasyfikatora zgłoszeń mieszkańców według pilności) albo generowanie insightów statystycznych (np. mapy cieplnej problemów w mieście). Tutaj powstają *produkty danych średniego szczebla* – modele predykcyjne lub raporty analityczne.

5. **Produkty danych premium**: Ostatni etap to przekucie wyników w konkretne *produkty wartościowe* dla odbiorców końcowych. Przykładowo: - **Usługi Data-as-a-Service**: Miasto może udostępniać pewne dane lub modele innym podmiotom na zasadzie subskrypcji. Np. model przewidujący obciążenie komunikacji miejskiej (wytrenowany na danych społecznych) może być oferowany lokalnym firmom transportowym poprzez API za opłatą abonamentową.

- **Licencje i protokoły**: Wypracowane w labie procedury (np. protokół angażowania mieszkańców w konsultacje z użyciem AI) mogą być licencjonowane innym samorządom jako pakiet know-how. To analogia do licencjonowania modeli AI – np. **OpenAI licencjonuje wytrenowane modele językowe**

partnerom, czerpiąc z tego opłaty ¹⁴, tutaj laboratorium może licencjonować „model społeczny” współpracy z AI.

- **Edukacyjne tokeny czasowe:** Ciekawym konceptem są *time-based coins* – np. mieszkańcy za udział w projektach otrzymują „żetony czasu”, które mogą wymienić na konsultacje eksperckie lub czas korzystania z infrastruktury labu. Choć nie jest to sprzedaż danych wprost, tworzy wewnętrzną gospodarkę wartości danych i czasu. W szerszym ujęciu, *time tokens* mogłyby być też mechanizmem monetyzacji – firmy płacą za określony czas dostępu do zespołu labu czy społeczności (coś na kształt crowdfundingu czasu).

- **Raporty i rekomendacje:** Ostatecznym „produktem premium” są często raporty zawierające rekomendacje polityk lub wdrożeń, poparte zebranymi danymi i analizami AI. Taki raport może być sprzedany (np. instytucjom centralnym) lub wykorzystany do pozyskania kolejnych grantów – jest to *skoncentrowana wiedza*, będąca efektem destylacji danych surowych przez proces labu (co przypomina **data distillation**, gdzie z surowych danych tworzy się zwarte, wartościowe informacje ¹⁵).

Przepływ danych w Social-AI Lab ma charakter cykliczny i zwrotny. Gdy produkty premium zostają wykorzystane (np. wdrożono nową usługę według rekomendacji), generują **nowe dane zwrotne** – np. mierniki korzystania z usługi, opinie mieszkańców. Dane te zasilają ponownie etap surowy, inicjując następną pętlę doskonalenia. Dzięki temu habitat działa niczym *ekosystem regeneracyjny danych*: dane raz zebrane krążą i przekształcają się, zamiast być zużyte jednokrotnie.

Krytyczne heurystyki sukcesu

Aby Laboratorium Społecznej SI utrzymało skuteczność i wartość swoich danych, musi stosować pewne **heurystyki zarządcze** – swoiste reguły kciuka, wypracowane na podstawie doświadczeń:

- **Utrzymuj rzadkość i unikatowość danych:** W dobie Big Data pokusą jest zbierać wszystko. Jednak lab koncentruje się na danych, które są *rzadkie lub wyjątkowe* – np. autentyczne opinie mieszkańców (czego nie znajdzie się w ogólnych zbiorach internetowych) lub dane z czujników specyficznych dla lokalnej społeczności. Ta *strategia rzadkości* oznacza też, że laboratorium nie rozdaje na prawo i lewo wszystkich surowych danych – publikuje raczej zagregowane wnioski, a surowe dane udostępnia warunkowo. Dzięki temu **utrzymuje wysoką wartość informacyjną**: dane nie dewaluują się przez nadmiar. Innymi słowy, lab dba o to, by nie doszło do „*saturacji rynku danych*” – gdy zbyt wiele podmiotów ma te same dane, ich wartość spada. Przykładowo, jeśli Social-AI Lab stworzy unikatowy zbiór nagrań interakcji seniorów z asystentem głosowym, to zamiast publikować całość za darmo, może udostępnić próbkę do celów naukowych, a pełny zestaw na licencji – zachowuje kontrolę nad podażą.
- **Unikaj pętli długu technologicznego:** Projekty są tak prowadzone, by nie generować długotrwałych zobowiązań bez pokrycia. Jeśli dany prototyp wymaga stałego nadzoru lub aktualizacji, lab stawia warunek przekazania go docelowemu właścicielowi (np. miejskiej jednostce) wraz z zasobami lub planem utrzymania. W przeszłości wiele inicjatyw smart city cierpiało na *technological debt loop* – technologie wdrożone pilotowo nie były później aktualizowane, co wymuszało kosztowne naprawy. Lab uczy się na tych błędach i np. stosuje **etapowe „bramki” (gating)**: projekt przechodzi dalej (np. z fazy prototypu do fazy wdrożenia), tylko gdy zapewniono środki i kompetencje na utrzymanie. To podejście naśladuje mechanizmy SBIR – **najpierw faza I (proof-of-concept za ~300 tys. \$), potem dopiero większa inwestycja** ¹⁶ – aby nie utopić dużych środków w niezwyfikowanym pomysle.
- **Filtruj nadmiar nieprzydatnych danych:** Lab nieustannie ocenia, które dane są faktycznie użyteczne. Tworzy polityki **data governance**, by nie gromadzić danych „na zapas” bez pomysłu na ich użycie. Po każdej rundzie projektu dokonuje się przeglądu: które zbiory danych dały wartościowe insighty, a które okazały się szumem? Te drugie są albo usuwane, albo odkładane z odpowiednim oznaczeniem. W myśl zasady FAIR, dane muszą być też **łatwo znajdowalne** –

dlatego nie przytłacza się repozytorium tysiącami plików, tylko utrzymuje *skondensowane magazyny danych* z jasnym katalogiem. Minimalizm ten służy temu, by zespół nie ugrzązł w bagnie nieprzeanalizowanych informacji. Lepiej mieć 10 kluczowych datasetów niż 1000 niesprawdzonych.

- **Ewoluuje benchmarki i cele:** Sukces laboratorium zależy od ciągłego podnoszenia poprzeczki. Gdy jakiś problem zostanie rozwiązany (np. model AI osiągnął 95% trafności w klasyfikacji zgłoszeń), to staje się on *nową normą*, a lab wyznacza kolejny cel (np. łączyć klasyfikację z przewidywaniem, jakie działania dają największą satysfakcję mieszkańców). **Ewolucja benchmarków** zapobiega stagnacji – nie można spocząć na laurach, bo dynamicznie zmienia się zarówno technologia, jak i potrzeby społeczne. Jest to analogiczne do praktyki w świecie AI: gdy modele maszynowego uczenia zaczęły bezbłędnie rozpoznawać obrazki w zbiorze MNIST, społeczność badawcza przeniosła uwagę na trudniejsze ImageNet, a gdy i tam osiągnięto ludzki poziom – definicja sukcesu poszła dalej (np. do ImageNet w warunkach ograniczonych zasobów). Laboratorium społeczno-SI musi podobnie redefiniować wyzwania, aby dane, które produkuje, pozostawały **rzadkie i wartościowe**.

Podsumowując, Laboratorium Społecznej SI to **żywy, uczący się organizm** nastawiony na zrównoważone współdziałanie ludzi i AI. Jego funkcjonowanie opiera się na subtelnym balansie: z jednej strony maksymalnie wykorzystuje zasoby lokalne (kompetencje ludzkie, dane, infrastrukturę), z drugiej – chroni unikatową wartość generowanych danych i rozwiązań. Dzięki temu modelowi społeczność zyskuje **korzyści informacyjne i społeczne**, a jednocześnie przygotowuje grunt pod dalszy rozwój gospodarki opartej na danych w sposób inkluzywny i etyczny.

Ekosystem 2: AI-ops jako usługa (AI-Ops-as-a-Service)

Definicja i założenia modelu

Model **AI-ops-as-a-service** (operacje SI jako usługa) przedstawia ekosystem, w którym habitat Human-AI działa jak „**zakład usługowy AI**” – świadczy na rzecz klientów zewnętrznych usługi oparte na sztucznej inteligencji, analogicznie do tego, jak data center oferuje moc obliczeniową w chmurze. W tym habitatcie zintegrowane są zaawansowane platformy AI (modele, infrastruktura obliczeniowa) wraz z zespołem specjalistów, a całość funkcjonuje ciągle, by **realizować zautomatyzowane zadania na zlecenie**. Można to porównać do „**AI w pudełku**” – zamkniętego środowiska, do którego wpływają dane i zapytania od klientów, a wypływają gotowe odpowiedzi, analizy czy decyzje.

Założenia modelu AI-ops-as-a-service są następujące: - Habitat ma charakter mocno **skomputeryzowany i zautomatyzowany**. W przeciwieństwie do laboratorium społecznego, tu celem jest minimalizacja bezpośredniej interakcji z ludźmi przy realizacji typowych zadań – ludzie są obecni głównie w rolach nadzoru, inżynierii i wsparcia, natomiast codzienne *operacje* (obsługa zapytań, przetwarzanie danych, monitorowanie) wykonuje AI. - Usługi mogą obejmować np. **przetwarzanie strumieni danych** (analiza wideo z satelitów, logów systemów IT – stąd określenie *AIOps* wywodzi się również z automatyzacji operacji IT), **utrzymywanie modeli AI dla klientów**, generowanie treści (np. tekstów, kodu) czy też podejmowanie decyzji w czasie rzeczywistym (jak autotrading na giełdzie). Klienci nie muszą budować własnej infrastruktury AI – zamiast tego „wynajmują” habitat, by wykonał pracę za nich. - Ekosystem jest zaprojektowany **modułowo**: posiada różne „kompetencje AI” w formie modułów, które mogą być zestawiane pod konkretne zlecenie. Na przykład moduł przetwarzania języka (LLM), moduł wizji komputerowej, moduł optymalizacji – w zależności od zadania (np. monitoring mediów społecznościowych, analiza obrazów z dronów) habitat aktywuje odpowiedni zestaw. - Model zakłada *nieprzerwane działanie*, często 24/7, i wysoką niezawodność. W związku z tym przyjęto wiele praktyk z tradycyjnych centrum danych i chmury: redundancję (nadmiarowe serwery), automatyczne przełączanie awaryjne, oraz **ciągłe uczenie maszyn** – systemy uczą się w trakcie pracy z napływającymi danymi, by

utrzymać lub poprawić jakość usług. W pewnym sensie habitat ten można wyobrazić sobie jako *kosmiczną stację przetwarzania danych*, być może nawet dosłownie umieszczoną na orbicie czy Księżycu w przyszłości, co nie jest fantazją – chiński startup już testuje modele AI działające na konstelacji satelitów na orbicie okołoziemskiej ¹⁷. - Założeniem organizacyjnym jest także **wysoka skalowalność usług**. Habitat AI-ops jest przygotowany, by dynamicznie skalować swoje zasoby w zależności od obciążenia – np. w godzinach szczytu uruchamia więcej instancji modelu, w nocy je wygasa. To wymaga zaawansowanego orkiestratora (który sam jest często systemem AI, stąd meta-poziom: AI zarządza innymi AI).

Przykładem realnego wdrożenia analogicznej idei jest komercyjna działalność takich firm jak **OpenAI** czy dostawcy chmurowi (AWS, Google Cloud) oferujący *MLaaS (Machine Learning as a Service)*. OpenAI ze swoim ekosystemem GPT funkcjonuje de facto jak czarna skrzynka AI-ops: klienci poprzez API przesyłają zapytania (np. tekst do wygenerowania podsumowania), a na serwerach OpenAI modele językowe wykonują zadanie i odsyłają wynik – wszystko w ułamkach sekund, na masową skalę. Habitat ten jest „zamknięty” dla użytkownika (nie wie on dokładnie, co dzieje się w środku), a rolą ludzi wewnątrz jest utrzymanie i doskonalenie systemu.

Założenie „*AI jako usługa użyteczności publicznej*” staje się coraz bardziej wymowne, bo popyt na takie usługi rośnie wykładniczo. Władze OpenAI opisują swój model biznesowy jako wielowymiarową układankę: od subskrypcji konsumenckich, przez usługi SaaS dla firm, po plany licencjonowania wyników działań AI w modelu success fee (np. procent od zysków klienta) ¹⁴ ¹⁸. Wszystko to wskazuje na kształtowanie się standardu, gdzie AI-ops-as-a-service będzie powszechnym elementem infrastruktury – jak dziś sieć elektryczna czy internet.

Źródła wartości ekonomicznej

W modelu AI-ops jako usługa wartości ekonomiczne mają charakter bardzo bezpośredni, bo usługi są oferowane na rynku. Główne **źródła przychodu i wartości** to:

- **Przetwarzanie danych i informacje na żądanie:** Habitat generuje przychody poprzez **opłaty za wykonane zadania AI**. Typowy mechanizm to *model użytkowy* – klienci płacą za jednostkę zużytego zasobu (np. **kredyty za tokeny w przypadku modeli językowych** czy za godzinę obliczeń GPU). OpenAI np. sprzedaje dostęp do GPT-4 za stawki od liczby przetworzonych tzw. tokenów (słów) w zapytaniu i odpowiedzi. Im więcej zapytań i im bardziej złożone, tym większy przychód. W 2024 r. firma ta notowała **przychód rządu 3,4 mld \$ rocznie** głównie z takich opłat za API i subskrypcji premium ¹⁹. To pokazuje skalę – dobrze działający habitat AI-ops może obsłużyć miliony mikropłatności dziennie.
- **Insighty i analizy sprzedawane jako produkt:** Oprócz przetwarzania surowych zapytań, ekosystem może oferować wyższy poziom wartości – gotowe *insighty*. Przykładowo, **Maxar** (dostawca danych satelitarnych) nie sprzedaje już tylko zdjęć, ale uruchomił usługę *Persistent Monitoring*, gdzie dostarcza klientom **wykryte zmiany i przewidywane zdarzenia** obserwowane na zdjęciach, przez aplikację Insights UI ²⁰. To dojrzały produkt: klient płaci za regularne raporty (np. ile nowych budynków powstało na danym obszarze w tym tygodniu), które są generowane automatycznie przez AI analizujące strumień danych. Źródłem wartości jest tu *aktualna wiedza*, której uzyskanie tradycyjnymi metodami byłoby wolniejsze lub droższe.
- **Benchmarking i porównanie modeli:** Habitat może też wytwarzać wartość oferując usługi **benchmarkingu AI** – np. porównuje modele klienta z własnymi na standardowych zadaniach. MLCommons ze swoimi testami MLPerf czy inicjatywy typu HuggingFace leaderboards to załączki takiej działalności. Przedsiębiorstwa mogą płacić za certyfikację, że ich model osiągnął określony wynik w standardowym teście (co bywa kluczowe marketingowo i regulacyjnie).
- **Protokoły i „AI-ops-as-a-service” packages:** Z czasem habitat buduje własne *know-how operacyjne* (jak optymalnie trenować modele, jak zarządzać drenażem mocy obliczeniowej, jak zapewniać ciągłość przy aktualizacjach). To know-how może samo w sobie stać się produktem – np. oferowanie innym

podmiotom licencji na oprogramowanie zarządzające AI-ops (sprzedaż platformy orkiestracji AI, wzorowanej na tej używanej wewnątrz). Inną formą są **usługi konsultingowe** – eksperci habitatu pomagają klientom zaadaptować wyniki działania AI do ich procesów biznesowych (np. integracja wygenerowanych insightów z systemem ERP firmy).

- **Uczące się kompetencje (model tuning on demand):** Habitat może osiągać przychód przez dostarczanie nie tylko odpowiedzi, ale *poprawy modeli klientów*. Przykład: firma farmaceutyczna korzysta z AI-ops, by przeszukać miliardy cząsteczek w poszukiwaniu kandydata na lek. Habitat wykonuje to zadanie, ale też uczy swój model chemiczny na danych klienta – powstały model, udoskonalony dzięki nowym danym, staje się wspólnym aktywem. Model można potem licencjonować dalej (za zgodą stron). OpenAI rozważa właśnie **modele licencyjne powiązane z wynikami klientów – np. udział w zyskach z odkrycia leku dokonanego z pomocą ich AI** ¹⁴. To nowy model biznesowy: AI-ops dostaje *royalty* za sukces wygenerowany u klienta, czyli współdzieli wartość dodaną.

Sumarycznie, źródłami wartości w tym ekosystemie są w głównej mierze *dane przetworzone w wiedzę lub decyzje*, sprzedawane w modelu usługowym: od surowej mocy predykcyjnej (płać-per-zapytanie) po wysublimowane produkty analityczne (raporty, modele, certyfikacje). To czyni AI-ops-as-a-service jedną z najbardziej bezpośrednio dochodowych form ekosystemu Human-AI – tu **dane są monetą** niemal dosłownie, bo każda jednostka informacji przetworzonej przez habitat ma swoją cenę.

Rachunek opłacalności (CAPEX, OPEX, ryzyka, punkty równowagi)

Model AI-ops-as-a-service jest **kapitałochłonny** na starcie, a także wymaga znaczących nakładów operacyjnych – choć przy odpowiedniej skali zyski mogą te koszty wielokrotnie przewyższyć.

CAPEX: Główne wydatki inwestycyjne dotyczą infrastruktury technicznej. Potrzebne są **centra danych o wysokiej wydajności**: setki lub tysiące procesorów GPU/TPU, szybkich serwerów, magazynów danych o dużej przepustowości. Dla przykładu, OpenAI aby wytrenować GPT-4 i świadczyć usługi, zawiązało partnerstwo z Microsoftem obejmujące inwestycję **\$1 mld już w 2019** w infrastrukturę i późniejsze transze – do 2023 Microsoft zainwestował w sumie ponad **\$13 mld** w ekosystem OpenAI ²¹. To pokazuje rząd wielkości CAPEX dla światowej klasy habitatów AI-ops. Oczywiście mniejsze skale też są możliwe (np. startupy AIOps z kilkudziesięcioma serwerami), ale aby obsłużyć globalne zapotrzebowanie, inwestycje liczone są w setkach milionów lub miliardach dolarów. CAPEX obejmuje też rozwój zaawansowanych modeli AI – samo trenowanie modelu jak GPT-4 szacunkowo kosztowało dziesiątki milionów USD, co można traktować jako nakład na "know-how" (model to kapitał intelektualny habitatu). Wreszcie, pewien CAPEX to stworzenie zaplecza oprogramowania: platformy orkiestrującej, interfejsów API, zabezpieczeń. Te koszty są duże, ale w modelu usługowym z reguły ponoszone **przed osiągnięciem przychodów**, stąd wymagane jest finansowanie (venture capital lub partnerstwa strategiczne).

OPEX: Operacyjnie habitat AI-ops jest także drogi. Największym komponentem jest **koszt energii i utrzymania serwerów**. Szacuje się, że sama obsługa ChatGPT (dziesiątek milionów zapytań dziennie) może kosztować **ok. \$700 tys. dziennie** w zużyciu mocy obliczeniowej ²². To obejmuje prąd dla GPU oraz chłodzenie centrów danych – co istotne, koszty te rosną niemal liniowo z liczbą zapytań. Do tego dochodzą koszty łączności sieciowych (transfery danych), amortyzacja sprzętu (GPU często trzeba wymieniać co 2-3 lata by pozostać konkurencyjnym), oraz płace wysoko wykwalifikowanej kadry (inżynierowie, DevOps, specjaliści od bezpieczeństwa). OpenAI zatrudnia kilkaset osób, a wynagrodzenia w branży AI są bardzo wysokie – analizy wskazują, że **koszty pracownicze mogą być porównywalne lub wyższe niż koszty samego sprzętu** ²³. Kolejnym elementem OPEX specyficznym dla AI-ops jest **koszt danych i ich anotacji** – aby utrzymać modele w formie, trzeba je okresowo dalej trenować na nowych danych, czasem kupować specjalistyczne zbiory lub zatrudniać ludzi do anotacji (np. do RLHF zatrudniano etatowo labelerów oceniających odpowiedzi modeli).

Ryzyka: W modelu AI-ops największe ryzyko to *niedopasowanie kosztów do przychodów*. Jeśli popyt nie osiągnie zakładanego poziomu, drogie centrum danych może świecić pustkami, generując stratę. Z drugiej strony, gdy popyt gwałtownie rośnie – ryzykiem staje się brak mocy przerobowych i spadek jakości usług (klienci odchodzą). Sytuacja OpenAI pokazuje takie napięcie: popyt na ChatGPT jest tak wielki, że **limitującym czynnikiem stała się dostępna moc obliczeniowa, a to wymuszało poszukiwanie kolejnych przychodów na zakup sprzętu** ²⁴. Innym ryzykiem jest **technologiczna utrata przewagi**: konkurencja też buduje modele i może oferować tańsze usługi. Ponieważ koszty zmienne (energia, hardware) są podobne dla graczy, kluczowe staje się mieć *lepszego modela lub tańszy dostęp do kapitału*. Istnieje też ryzyko regulacyjne – np. wprowadzenie wymogów prawnych (jak europejskie AI Act) może zwiększyć koszty compliance (certyfikacje, audyty) lub ograniczyć niektóre zastosowania komercyjne AI.

Punkt równowagi (break-even): Ze względu na wysokie koszty stałe, ekosystem AI-ops potrzebuje dużej skali, by osiągnąć break-even. Szacunkowo, jeżeli dzienny koszt operacji to \$700k ²², roczne to ok. \$255 mln. Jeśli doliczymy amortyzację inwestycji i koszty pracowników, można założyć, że roczny koszt operacyjny dużego habitatu to np. \$500 mln. Aby to pokryć, przy marży 50%, przychód roczny musiałby wynosić \$1 mld. Oznacza to konieczność obsługi bardzo wielu klientów i zapytań. OpenAI planowało na 2023 przychód ~\$200 mln, a na 2024 już \$1 mld – co pokazuje tempo, w jakim muszą rosnąć przychody, by zbliżyć się do rentowności ²⁵. Tego typu habitaty często przez pierwsze lata działają na deficycie (subsydiowanym inwestycjami), by dopiero po osiągnięciu odpowiedniej skali wyjść na plus. W dokumentach NASA dot. opłacalności R&D wspomina się, że **wystarczająca skala potrafi uczynić kosmiczne (dosłownie) projekty opłacalnymi** – np. **ISS National Lab odnotowuje coraz większe zaangażowanie przemysłu, co sugeruje, że przy spadku kosztów wynoszenia i rosnącej liczbie eksperymentów, ROI staje się atrakcyjne** ^{26 27}. Analogicznie w AI-ops: spadek kosztów jednostkowych (np. dzięki nowszym chipom, optymalizacjom) i wzrost obrotów tworzą punkt, gdzie krzywe się przetną.

Z perspektywy **ROI i NPV** inwestorów – model ten jest wysoki ryzyka, wysokiej nagrody. Jeśli habitat zdobędzie *quasi-monopol* lub wczesną dominację, generuje przepływy pieniężne ogromne (monetyzacja każdej interakcji użytkowników z AI). Ale wartość obecna netto projektu może być ujemna, jeśli dyskonto inwestycji i prawdopodobieństwo sukcesu są niekorzystne. Dlatego wielu graczy wchodzi w partnerstwa (np. Microsoft+OpenAI) dzieląc ryzyko i przyszłe zyski. W scenariuszu sukcesu, **przewaga pierwszego ruchu** daje efekt sieciowy – więcej danych klientów poprawia modele, lepsze modele przyciągają więcej klientów – co umacnia pozycję lidera (możliwe powstanie naturalnego duopolu czy oligopolu AI-ops globalnie).

Mechanizmy redukcji kosztów i maksymalizacji wartości

W tak wymagającym finansowo ekosystemie kluczowe jest stosowanie zaawansowanych mechanizmów technologicznych i organizacyjnych, aby **obniżyć koszty jednostkowe i zwiększać wartość dodaną**. Oto najważniejsze z nich:

- **Zaawansowana optymalizacja i ekologia obliczeń:** Habitat AI-ops dąży do *maksymalnego wykorzystania zasobów*. Jednym z mechanizmów jest optymalizacja modeli pod kątem wydajności – np. używanie pruned models, quantization (kwantyzacja parametrów), kompilatory optymalizujące sieci neuronowe na dany hardware. Ponadto stosuje się **harmonogramowanie zadań**: nie wszystkie zadania muszą być wykonane natychmiast, więc zadania niższego priorytetu wykonuje się w dolinach obciążenia (to analogia do *demand shaping* w energetyce). Habitat może też dynamicznie *wybierać najtańszą ścieżkę obliczeń* – np. jeśli odpowiedź może wygenerować mniejszy model za 1/10 kosztu w nieco dłuższym czasie, a użytkownik nie potrzebuje natychmiast, to system tak robi. Te decyzje często podejmuje sama AI, analizując

metryki użytkownika. W efekcie ekosystem przypomina **ekosystem naturalny, gdzie nic się nie marnuje**: niewykorzystana moc jest alokowana np. na trenowanie nowych modeli (by nie chodziły serwery na jałowo).

- **Autonomia i samonaprawa:** W duchu automatyzacji, habitat implementuje koncepcje **self-* (self-healing, self-scaling, self-optimizing)**. Awarie sprzętu są automatycznie wykrywane i obciążenie przełączane na inne węzły. Systemy monitoringu (nierzadko z algorytmami ML) przewidują, kiedy serwer może ulec awarii (np. na podstawie temperatur, błędów) – i wyłączają go prewencyjnie do serwisu. Takie *autonomiczne operacje* zmniejszają koszty przestojów i interwencji ludzkich. Zespoły ludzkie mogą być przez to mniejsze, co redukuje OPEX. Co więcej, automatyzacja obejmuje też **cykle uczenia modeli** – tzw. *MLOps*. Modele mogą być trenowane/podnoszone do nowszych wersji w dużej mierze bez ręcznej pracy (pipelines Ci/CD dla modeli).
- **Pętla danych i uczenie ciągłe:** Każda interakcja klienta z systemem staje się potencjalnie *danymi do dalszego uczenia*. Na przykład pytania zadawane do asystenta AI ujawniają, jakich informacji szukają klienci – habitat może trenować swoje modele FAQ na tych zapytaniach, by następnym razem odpowiadać szybciej lub offline. Błędy, jakie popełni AI, są logowane i trafiają do kolejnego cyklu treningowego jako *przykłady do unikania*. To tworzy mechanizm **ciągłego doskonalenia**: im dłużej habitat działa, tym jest mądrzejszy i tym większą wartość może oferować (np. szybsze odpowiedzi, nowa funkcjonalność), bez proporcjonalnego wzrostu kosztów – bo uczy się we własnym trybie (często nocą, gdy jest mniejsze obciążenie). Ten *feedback loop* to kręgosłup przewagi – podobnie jak Google’owi udało się zdominować rynek wyszukiwarek dzięki miliardom zapytań uczącym algorytm rankingu, tak habitat AI-ops broni się danymi, które sam generuje i zbiera przy obsłudze klientów.
- **Cyfrowe feromony w orkiestracji zadań:** W rozproszonej infrastrukturze AI-ops do koordynacji wykorzystuje się idee *stigmergii*. Poszczególne komponenty systemu zostawiają „ślady” o swoim stanie i preferencjach w wspólnym środowisku (np. w rejestrach, w systemie kolejowania zadań). Te **cyfrowe feromony** to np. wpis „GPU_23 przeciążony, zadania graficzne omijać” albo „Model X osiągnął limit błędów – potrzebna walidacja”. Inne moduły odczytują te sygnały i dostosowują zachowanie – np. scheduler odciąża GPU_23, a moduł walidacyjny przyspiesza testy modelu X. Dzieje się to bez centralnego nadzoru, podobnie jak **mrówki koordynują się zostawiając ślady feromonowe** ¹⁰. Taki zdecentralizowany mechanizm poprawia skalowalność (system może mieć setki modułów i węzłów, a jednak dzięki wymianie sygnałów działa spójnie) i odporność (awaria jednego elementu jest wykrywana przez brak jego „zapachu” i inne moduły to kompensują). W efekcie koszty przestojów maleją, a **efektywność wykorzystania zasobów rośnie** – system sam „wie”, gdzie jest moc niewykorzystana i tam kieruje nowe zadania.
- **Kompetencje AI jako produkt pakietowy:** Habitat zwiększa wartość oferowaną klientom przez łączenie kompetencji. Zamiast sprzedawać pojedyncze modele, tworzy *pakiety rozwiązań*. Np. dla branży e-commerce: pakiet AI-ops obsługujący całość operacji AI sklepu – od rekomendacji produktów, przez obsługę czatu z klientem, po optymalizację logistyki magazynu. Dzięki temu klient płaci subskrypcję za całość (co zwiększa przychód), a habitat może wewnętrznie **synergicznie wykorzystać dane z różnych modułów** (np. dane z czatu mogą zasilić model rekomendacji, by lepiej personalizować oferty – to zwiększa skuteczność, więc i wartość usługi). Ta synergia jest możliwa dzięki centralizacji AI-ops – działając jak *hub*, widzi przekrojowo procesy, co pojedynczym rozproszonym systemom u klienta byłoby trudne. W ekonomii to oznacza *efekty skali i zakresu*: koszt obsługi wielu funkcji razem jest niższy niż osobno, a wartość dla klienta – wyższa. Habitat staje się w ten sposób nie tylko dostawcą mocy obliczeniowej, ale **partnerem biznesowym** wspierającym wiele aspektów działalności klientów (co buduje lojalność i stabilność przychodów).

Podsumowując, AI-ops-as-a-service wdraża najbardziej zaawansowane praktyki „**inteligentnej automatyzacji**” zarówno wewnątrz (aby obniżyć koszty), jak i na zewnątrz (aby zaoferować unikatową wartość). Kombinacja **autonomii, ciągłego uczenia i architektury inspirowanej systemami organicznymi** (stigmergia, pętle zwrotne) czyni z tego habitat swoisty *organizm cyfrowy*, zdolny adaptować się i optymalizować praktycznie bez przerwy. Jest to konieczne, bo marginesy opłacalności bywają wąskie, a wymagania jakościowe – wysokie. Dzięki tym mechanizmom najlepszym graczom na rynku udaje się stopniowo je poszerzać – np. OpenAI zdołało istotnie obniżyć koszty generowania tekstu z GPT-3 do GPT-4 poprzez usprawnienia modelu i infrastruktury (efekt skali pozwolił nawet zaoferować darmowy ChatGPT szerokiej publiczności, finansowany wpływami od użytkowników płacących i partnerów). Innymi słowy, „**cyfrowa ekologia**” AI-ops to nieustanne poszukiwanie równowagi między wydajnością a kosztem, gdzie każdy element systemu jest zarówno konsumentem, jak i dostawcą informacji pomagających tę równowagę utrzymać.

Rola człowieka w pętli (HITL)

Choć model AI-ops-as-a-service dąży do maksymalnej automatyzacji, **człowiek nadal pełni w nim krytyczne role**, w nieco inny sposób niż w labie społecznościowym:

- **Projektowanie i nadzór strategiczny:** Na etapie tworzenia usług to ludzie decydują, jakie **kryteria jakości i bezpieczeństwa** muszą spełniać systemy AI. Inżynierowie i menedżerowie definiują np. że model czatbotowy nie może przekroczyć określonego wskaźnika toksyczności odpowiedzi. Człowiek ustala także *granice autonomii* – np. które decyzje system może podejmować sam (restart serwerów), a gdzie wymagana jest autoryzacja (np. wdrożenie nowego modelu do produkcji może wymagać przeglądu człowieka). W ten sposób ludzie ustanawiają **reguły meta-poziomu** działania habitatu.

- **Interwencje eksperckie (expert-in-the-loop):** W codziennym działaniu większość zapytań obsługuje AI samodzielnie, ale pewna klasa trudnych lub nietypowych przypadków może być *eskalowana do człowieka*. Na przykład, jeśli system wykryje bardzo niestandardowe zapytanie klienta, którego nie jest pewien, może przekazać je do analityka (to analogia do wsparcia 2. linii w call center). Człowiek rozwiązuje problem i jednocześnie uczy system, jak radzić sobie z podobnym w przyszłości. Ten mechanizm **Human-in-the-Loop** jest często stosowany w systemach ML w produkcji – np. moderacja treści: model oznacza potencjalnie wrażliwy content, ludzie go przeglądają i decyzja (ban/ok) wraca do modelu jako nowy przykład do nauki.

- **Poprawa jakości modelu przez RLHF:** Wysoką wartość przynosi nadal *ludzka ocena jakości* odpowiedzi modeli. **Reinforcement Learning from Human Feedback (RLHF)** okazał się przełomowy w dostrajaniu modeli językowych – ludzie oceniając i rankingując odpowiedzi modelu umożliwili stworzenie GPT, który bardziej słucha poleceń i jest mniej toksyczny ¹¹ ²⁸. W habitatcie AI-ops takie procesy są stale w toku: zatrudnia się zespoły labelerów i testerów (czasem rozproszonych globalnie), którzy **ciągle karmią system informacją zwrotną**. Ich praca bezpośrednio zwiększa wartość oferowaną klientom – np. sprawia, że odpowiedzi asystenta są poprawniejsze i bardziej pomocne. Co więcej, okazuje się, że zaangażowanie ludzi może nawet pozwolić mniejszemu modelowi przewyższyć większy niekontrolowany – **w testach OpenAI model 1.3 mld z RLHF był preferowany od 175 mld parametrowego GPT-3 bez RLHF** ²⁸, co pokazuje ogromną dźwignię jakościową dzięki ludzkim ocenom.

- **Inicjowanie nowych usług i innowacji:** Ludzie – zwłaszcza analitycy biznesowi i naukowcy danych – obserwując interakcje systemu z klientami, mogą dostrzec **nowe potrzeby lub możliwości**, których sama AI nie „zrozumie” (bo skupia się na tym, do czego ją nauczono). Przykładowo, operatorzy AI-ops mogą zauważyć, że wielu klientów zadaje pytania o określony temat X, który nie jest dobrze obsługiwany. To impuls do opracowania nowego modułu lub rozszerzenia usługi. Człowiek więc pełni rolę *kreatora i katalizatora zmiany*. Nawet w wysoce autonomicznych środowiskach kosmicznych (jak planowane bazy księżycowe) przewiduje się, że „**człowiek w pętli**” będzie inicjował adaptacje systemu, gdy warunki się zmieniają lub pojawiają się nowe cele misji, bo ludzka intuicja i zdolność do abstrakcji wykracza poza bieżące algorytmy.

Z perspektywy roli HITL w AI-ops można zauważyć, że *ludzie przesuwają się tutaj bardziej „w górę” abstrakcji*: mniej ręcznej obsługi poszczególnych transakcji (to robi AI), a więcej roli nadzorcy, nauczyciela i menedżera systemu. Zamiast gasić każdy pożar, człowiek ustawia strażakom (AI) procedury i sam interweniuje przy najtrudniejszych pożarach. Taka konfiguracja zwiększa skalowalność – jeden człowiek nadzoruje pracę wielu agentów AI. Wskaźniki **HITL performance** mierzy się tu np. jako *odsetek przypadków wymagających eskalacji do człowieka* (im niższy, tym system dojrzalszy), *czas reakcji człowieka* i *zgodność decyzji AI z decyzjami eksperta*. W praktyce, celem jest z czasem minimalizować konieczność interwencji ludzkiej bez utraty jakości – ale *nie do zera*, bo pewien poziom czynnika ludzkiego jest gwarantem kontroli (szczególnie w kontekście ryzyka błędnych decyzji AI).

Zatem, w AI-ops humans **increase value by injecting judgment, ethics, and continuous improvement**. Mają oko na to, by automaty nie zboczyły z wyznaczonych ram i by stale się uczyły właściwych zachowań. Można powiedzieć, że **człowiek jest tutaj trenerem i sędzią** – przygotowuje AI do meczu (poprzez szkolenie) i ocenia jej zagrania, korygując strategię, ale większość gry odbywa się autonomicznie. Ten model współpracy umożliwia osiągnięcie skali usług niemożliwej do obsłużenia czysto ludzkimi siłami, a jednocześnie zachowuje kluczowe atrybuty ludzkiej inteligencji w systemie.

Przepływ danych: od surowych danych do produktów premium

W habitatcie AI-ops przepływ danych jest sercem działalności – to wysoko zautomatyzowana linia produkcyjna informacji. Można ją opisać następująco:

1. **Wejście danych surowych:** Płyną one dwoma głównymi strumieniami:
2. **Dane od klientów (zadania):** Każde zlecenie od klienta (np. zapytanie do API, przesłany zbiór do analizy) to pakiet danych wejściowych. Może to być tekst, obraz, strumień wideo, liczby itp. Są one traktowane jako surowiec do przetworzenia.
3. **Dane środowiskowe:** Logi systemowe, metadane o obciążeniu, wyniki monitoringu działania modeli. One nie są „klientowskie”, ale stanowią surowe dane do wewnętrznej optymalizacji (np. zużycie GPU w czasie). W przeciwieństwie do tradycyjnych habitatów, tu surowe dane napływają **ciągłym strumieniem w czasie rzeczywistym** (real-time streaming). System musi być gotów je odebrać i od razu reagować (w wielu usługach liczy się np. opóźnienie poniżej 100 ms).
4. **Routing i pre-processing:** Nadchodzące dane są **klasyfikowane i kierowane** do odpowiednich modułów. Działa tu coś w rodzaju *inteligentnej rozdzielni*. Przykładowo, jeśli przychodzi zapytanie tekstowe do modelu językowego, front-end systemu od razu rozpoznaje typ (np. poprzez inspekcję formatu, tokenów) i kieruje je do klastra LLM. Jeżeli są to dane tabelaryczne z czujników – trafiają do modułu analizy szeregów czasowych itd. W ramach pre-processingu dane mogą być oczyszczone (np. usunięcie potencjalnie niebezpiecznych znaczników HTML z tekstu) oraz wzbogacone w metadane (np. identyfikator klienta, priorytet SLA, timestamp). Ten krok zapewnia, że dane **wejdą do „fabryki” w standaryzowanej formie** i z odpowiednią etykietą co do potrzebnego przetwarzania.
5. **Przetwarzanie rdzeniowe (core processing):** To serce przepływu – właściwe algorytmy AI dokonują **analizy, predykcji lub generacji**. W zależności od usługi, może tu nastąpić:
6. *Inferencja modelu ML:* np. sieć neuronowa generuje odpowiedź tekstową, klasyfikuje obraz, przewiduje anomalię w danych.
7. *Wykonanie złożonej procedury:* W usługach bardziej złożonych wiele modeli działa sekwencyjnie. Np. zapytanie głosowe: najpierw moduł rozpoznaje mowę do tekstu, potem LLM analizuje

intencję i generuje odpowiedź, na końcu moduł syntezy mowy odczytuje odpowiedź. To jest *pipeline*, w którym dane przechodzą przez kilka transformacji.

8. **Agregacja i analiza big data:** Niektóre zadania (np. raport dzienny) wymagają zebrania danych z wielu źródeł i syntezy (moduły typu map-reduce, zapytania do hurtowni danych). W każdym przypadku wynikiem tego etapu jest pewna **informacja wynikowa** – surowy produkt, np. wygenerowany tekst, wykryta lista obiektów na obrazie, tablica wyników analizy. Ten wynik bywa jeszcze w formie maszynowej, nie gotowej dla klienta (np. embeddingi, które dopiero posłużą do sformułowania odpowiedzi).
9. **Post-processing i walidacja:** Zanim wynik trafi do klienta, następuje **ostatnia kontrola jakości i ewentualne wzbogacenie**. Jeśli system jest dobrze zaprojektowany, większość odpowiedzi będzie poprawna, ale np. w module czatowym może być nałożony *filtr moderujący* – skanuje wygenerowany tekst pod kątem niedozwolonych treści i w razie potrzeby cenzuruje lub modyfikuje odpowiedź (to może być automatyczne lub jeśli system nie jest pewien, sygnał by dołączyć przeprosiny, że czegoś nie może odpowiedzieć). Często dodaje się również metadane do wyniku: np. **stopień pewności predykcji** albo unikalny ID zapytania do celów śledzenia. Jeśli w procesie brał udział człowiek (eskalacja), to na tym etapie efekt pracy człowieka (np. poprawiona odpowiedź) jest wprowadzany do wyniku. Ten etap stanowi *gating*, który chroni przed wypuszczeniem ewidentnie błędnego lub niebezpiecznego wyniku do odbiorcy (co mogłoby mieć konsekwencje prawne lub utratę klienta). Walidacja może być wielowarstwowa – np. OpenAI implementuje *systemowy prompt i polityki*, które modyfikują wyjście modelu zgodnie z wytycznymi (np. by unikał porad medycznych).
10. **Wyjście – dostarczenie produktu premium:** Ostatecznie przygotowana odpowiedź lub rezultat jest **dostarczany klientowi** poprzez odpowiedni kanał: API, interfejs aplikacji, email z raportem, itd. Z punktu widzenia klienta, jest to *produkt końcowy*, za który płaci – np. konkretny wygenerowany utwór graficzny, przefiltrowana informacja, wygodny raport PDF. W modelu subskrypcyjnym może to być też **ciągły strumień wyników** (jak w przypadku monitoringu – klienci otrzymują alerty lub mogą na żywo widzieć rezultaty w panelu). W tym sensie habitat może realizować **Data-as-a-Service** lub **Insights-as-a-Service** – wyniki są często podawane w formie gotowej do spożycia, co podnosi ich wartość. Dla przykładu, Maxar sprzedaje dostęp do platformy SecureWatch, gdzie klienci nie tylko przeglądają surowe zdjęcia satelitarne, ale dostają narzędzia analityczne i *na życzenie generowane statystyki* ²⁹. Tutaj habitat pełni funkcję zaplecza tej platformy.
11. **Logowanie i sprzężenie zwrotne:** Każda transakcja jest **logowana** – zarówno dla rozliczeń (np. zliczanie użycia do faktury, bądź spalanie tokenów), jak i dla potrzeb doskonalenia systemu. Log zawiera parametry wejściowe (w zanonimizowanej formie, by chronić dane klienta), decyzje, jakie podjęły poszczególne moduły, oraz wynik. Te logi są następnie analizowane offline przez system (lub ludzi) – szuka się błędów, ocenia czasy odpowiedzi, uczy modele na feedbacku. Na ich podstawie generowane są też **metryki SLA** (czy dotrzymano czasu odpowiedzi, czy dokładność była w normie, etc.). To zamyka pętlę: dane z wyjścia wracają jako *dane surowe wewnętrzne* do punktu 1 (np. w nocy system przetwarza dzienne logi, fine-tuninguje modele).
12. **Monetyzacja i licencjonowanie:** Równolegle do powyższego przepływu czysto technicznego, odbywa się przepływ związany z monetyzacją. Każdy wynik jest skojarzony z **modelem biznesowym**: albo został odjęty z przedpłaconego pakietu (subskrypcja), albo generuje event do systemu billingowego (naliczenie opłaty za użycie). W wypadku produktów premium, może wchodzić w grę **licencjonowanie wyników** – np. jeśli habitat wygeneruje dla klienta unikalny model lub zbiór danych (na jego zlecenie i danych), to ustala się licencję na ten produkt (czy

habitat może go używać dla innych, czy jest wyłączny dla klienta). Coraz częściej dyskutuje się też tokenizację dostępu: w przyszłości klient mógłby np. wykupić *NFT reprezentujące pewną ilość mocy AI* do wykorzystania, lub zarabiać tokeny dzieląc się swoimi danymi do trenowania (choć to raczej spekulatywne). Obecnie dominują tradycyjne modele: API w chmurze (płać za tysiąc wywołań) oraz subskrypcje czasowe (np. miesięczny abonament za określony pakiet usług). Ważne jest, że habitat ma **telemetrię** do szczegółowego naliczania – stąd w zestawieniu z danymi przepływu technicznego płyną też *dane biznesowe* (np. identyfikator klienta, taryfa).

Podsumowując, przepływ danych w AI-ops jest dobrze naoliwionym, wieloetapowym procesem, gdzie surowe dane klienta są szybko przekształcane przez szereg transformacji AI w gotową informację lub akcję, a nad wszystkim czuwa mechanizm walidacji i logowania zapewniający jakość i umożliwiający uczenie się z doświadczeń. W pewnym sensie to **cyfrowa fabryka**, działająca w czasie rzeczywistym, w której *dane to surowiec, modele to maszyny, a ludzie to inżynierowie-projektanci i kontrolerzy jakości*.

Krytyczne heurystyki sukcesu

Aby habitat AI-ops-as-a-service odniósł długoterminowy sukces, musi kierować się kilkoma kluczowymi heurystykami, które pozwalają utrzymać konkurencyjność i uniknąć typowych pułapek:

- **Zapewnij rzadkość poprzez przewagę jakości:** W świecie usług AI, jeśli pewna informacja czy funkcja staje się powszechna (każdy może ją łatwo wygenerować), jej wartość drastycznie spada. Dlatego habitat musi stale utrzymywać **przewagę jakościową lub unikatowe możliwości** – to jest jego *rzadkość*. Przykładowo, będąc jedynym dostawcą modelu o określonych umiejętnościach (np. najlepszego w kodowaniu, czy ekskluzywnego partnera mającego dostęp do jakichś danych), habitat unika wojny cenowej. Oznacza to m.in. stałe inwestycje w R&D (nie osiadać na laurach) i chronienie swoich najlepszych modeli przed klonowaniem (poprzez tajemnicę handlową lub przewagę danych treningowych). Innymi słowy – **utrzymuj technologiczną unikalność**.
- **Unikaj przetrenowania na wczorajszych danych (saturacja poznawcza):** Modele AI mogą popaść w stagnację, jeśli karmione są w kółko tymi samymi danymi lub optymalizowane wyłącznie pod dzisiejsze benchmarki. To grozi *utratą ogólności* – system staje się świetny w tym, co było mierzone, ale nie radzi sobie z nowościami. Aby temu zapobiec, habitat stosuje heurystykę: **zachowaj różnorodność wyzwań**. W praktyce oznacza to, że okresowo wprowadza się nowe rodzaje zadań testowych, nowe domeny, a stare benchmarki się aktualizuje lub porzuca po opanowaniu. Jak zauważa Stanford HAI, **dobry benchmark AI powinien ewoluować i być wzbogacany o nowe problemy, by uniknąć nasycenia** (saturation) i manipulacji ³⁰. Habitat musi czynić podobnie: stale szukać „dziury w całym” swoich modeli i karmić je nowymi bodźcami, nawet losowymi czy syntetycznymi (np. generować adversarial examples), by utrzymać je czujnymi i elastycznymi.
- **Monitoruj „pętlę długu technicznego” w infrastrukturze:** Złożone systemy mogą wpaść w spiralę tech debt – szybkie łatki i obejścia dodawane w pośpiechu (np. by usunąć bug natychmiast) z czasem tworzą kruchą strukturę trudną w utrzymaniu. Habitat musi działać wedle heurystyki: **regularnie refaktoryzuj i inwestuj w porządk**i. Choć kuszące jest ciągle dodawać nowe funkcje dla klientów, ważne jest, by co jakiś czas *splacać dług techniczny*: upraszczać architekturę, usuwać przestarzałe moduły, integrować powielone funkcjonalności. NASA przy rozwoju systemów kosmicznych stosuje podejście fazowe z przeglądami krytycznymi – zanim system przejdzie dalej, usuwa się wykryte błędy i niedoróbki. Podobnie habitat AI-ops powinien mieć wewnętrzne „gates” – np. co kwartał audyt architektury i priorytetyzacja refaktoryzacji. To

zapobiegnie sytuacji, że rosnący system staje się niewydolny i kruchy (co mogłoby doprowadzić do załamania usług i ucieczki klientów).

- **Filtruj bezużyteczne dane i interakcje:** Z perspektywy rozwoju modeli – nie wszystkie dane są tak samo wartościowe. Heurystyka tutaj to **skup się na danych, które poprawiają model, pozbądź się szumu**. Habitat powinien więc analizować, które zapytania klientów wносиły nową informację do uczenia (np. przypadki gdzie model był niepewny, lub gdzie nastąpiła korekta przez człowieka) – i na nich się koncentrować w treningu. Dane redundantne (powtarzające utarte wzorce) można sample'ować rzadziej lub pomijać, by nie przedłużać niepotrzebnie treningu i nie zagrazać pamięci. Tak samo w logach monitoringu – warto określić, jakie sygnały są istotne do analizy, a resztę archiwizować w sposób skompresowany (lub kasować po czasie). W dobie, gdy każda interakcja generuje gigabajty logów dziennie, *sztuką jest wyłowić perełki z hałasu*. Jest to zasada analogiczna do inżynierii danych w big data: lepiej mieć mniej danych o wysokiej jakości sygnału niż zalewać modele ogromem danych z niskim *information gain*.
- **Zapewnij transparentność i zaufanie (szczególnie dla premium klientów):** Dla utrzymania klientów – szczególnie w zastosowaniach krytycznych – heurystyką sukcesu jest **przejrzystość działania i stała kontrola wyników**. Oznacza to np. dostarczanie klientom raportów z jakości usług (dowody, że spełniamy SLA), udostępnianie interfejsów do audytu (np. klienci korporacyjni mogą wglądać w logi swoich zapytań, albo testować modele na własnych danych testowych), certyfikacje niezależnych jednostek (np. zgodność z normami ISO AI). Choć mogłoby się wydawać, że black-box, który „po prostu działa”, jest akceptowalny, w praktyce duzi klienci cenią sobie **poczucie kontroli**. Stąd np. OpenAI publikuje *System Card* GPT-4, opisujący jego ograniczenia i testy bezpieczeństwa ³¹, a dostawcy chmury często umożliwiają klientom zachowanie własności danych i modeli (np. szyfrowane instancje). Transparentność buduje zaufanie, co jest walutą nie mniej ważną niż sama moc modelu – a utrzymanie zaufania klientów pozwala przetrwać ewentualne wpadki (bo klienci wierzą, że dostawca gra fair i szybko naprawi).

Podsumowując, **sukces ekosystemu AI-ops** wymaga zarówno utrzymywania przewagi technologicznej i efektywnościowej, jak i mądrego zarządzania danymi i relacjami z klientami. To balans między byciem na **awangardzie innowacji** (ciągłe ulepszanie modeli, wprowadzanie nowych usług zanim zrobi to konkurencja), a **konserwatywną inżynierią niezawodności** (solidność, czystość architektury, zgodność z regulacjami). Te heurystyki – unikalność, różnorodność, spłacanie długu, filtrowanie szumu, transparentność – działają jak *wbudowane mechanizmy korekcyjne*, które utrzymują habitat na ścieżce wzrostu, zamiast pozwolić mu ugrzęznąć w problemach własnego sukcesu (przerost formy, zbytne uogólnienie, utrata zaufania).

(Analogiczne analizy dla pozostałych trzech ekosystemów – Prototypowni danych, Habitatu regeneracyjnego oraz Środowiska certyfikacji i benchmarków – znajdują się w kolejnych sekcjach raportu.)

Ekosystem 3: Prototypownia danych (Data Prototyping Lab)

Definicja i założenia modelu

Prototypownia danych to habitat skoncentrowany na **szybkim tworzeniu i testowaniu prototypów rozwiązań AI/ML**. W tym ekosystemie głównym produktem są *eksperymenty i dowody koncepcji (PoC)* – krótko trwające, iteracyjne projekty mające na celu zweryfikowanie nowych pomysłów z zakresu przetwarzania danych. Model zakłada, że habitat wyposażony jest w narzędzia (zarówno sprzęt, jak i oprogramowanie oraz zestawy danych), które pozwalają zespołom projektowym szybko budować prototypy modeli, algorytmów czy procesów analitycznych, a następnie testować je w warunkach

zbliżonych do docelowych. Prototypownię danych można porównać do **kosmicznego centrum eksperymentalnego** – analogicznie do np. stacji badawczych na Antarktydzie czy laboratoriów typu **Flight Opportunities** NASA, gdzie w krótkich lotach suborbitalnych testuje się nowe technologie w zredukowanej skali przed pełną misją kosmiczną ³². Tutaj natomiast testuje się nowatorskie pomysły na danych przed wdrożeniem w pełnej skali biznesowej.

Załoženiami modelu są: - **Szybkość i zwinność**: Każdy prototyp budowany jest w krótkim cyklu (tygodnie, góra kilka miesięcy) – celem jest *szybkie osiągnięcie wniosków*: czy dany pomysł ma szansę być wartościowy, czy nie. W tym celu prototypownia stosuje metodologię **agile, a wręcz hyper-agile** – np. hackathony 2-dniowe, sprinty 2-tygodniowe, programy akceleracyjne 3-miesięczne.

- **Niskie koszty początkowe i ograniczony CAPEX**: Zamiast budować od razu docelową infrastrukturę, korzysta się z współdzielonych zasobów i symulacji. Habitat może wykorzystywać np. *chmurę obliczeniową on-demand* do testów (żeby nie kupować sprzętu zanim nie zajdzie potrzeba), generować **syntetyczne dane** tam, gdzie brak prawdziwych, i stosować makiety interfejsów zamiast pełnych aplikacji. Idea to „**fail fast, fail cheap**” – porażka prototypu ma być tania i akceptowalna, a sukces dopiero potem uzasadnia większe inwestycje.

- **Środowisko zbliżone do docelowego**: Prototypownia stara się jak najlepiej *symulować warunki*, w jakich docelowo rozwiązanie ma działać, aby wiarygodnie ocenić jego przyszłą efektywność. Przykładowo, jeżeli prototypem jest model do analizy obrazów satelitarnych, habitat zapewni dostęp do choćby niewielkiego strumienia danych z prawdziwych satelitów lub użyje wysokiej jakości symulatora satelitarnych obrazów. Porównać to można do analogów misji marsjańskich na Ziemi (jak habitaty Mars Desert Research Station) – gdzie warunki marsjańskie są imitowane w zakresie możliwości. Tu warunki np. przemysłowe czy rynkowe są imitowane dla testu.

- **Dostęp do szerokiego zakresu kompetencji i metod**: W prototypowni funkcjonuje interdyscyplinarny zespół (data scientist, inżynier, specjalista dziedziny), często rotacyjny. Habitat ma charakter „**open lab**” – angażuje również partnerów zewnętrznych: startupy, studentów, kontraktorów – by w krótkim czasie złożyć zespół mający wszystkie potrzebne umiejętności pod konkretny prototyp. To ułatwia testowanie bardzo różnych pomysłów – np. jednego miesiąca prototypuje się model medyczny (więc zaprasza się lekarza i specjalistę od AI w medycynie), a kolejnego optymalizację logistyczną (więc potrzebny logistyk i programista algorytmów optymalizacyjnych).

- **Tolerancja ryzyka i uczenie się na błędach**: Kultura pracy prototypowni zakłada, że **większość prototypów może się nie udać** – i jest to normalne. Ważne, aby z każdego wyciągnąć lekcje. Dokumentuje się więc i pozytywne wyniki, i porażki, tworząc wewnętrzną bazę wiedzy *dlaczego coś nie zadziało*. To zapobiega powtarzaniu tych samych ślepych uliczek przez innych. NASA w programach lotów testowych (np. X-planes) miała motto "failure is not failure if we learn from it" – analogicznie tu.

W praktyce, realnymi odpowiednikami prototypowni danych są **działy R&D wielu firm technologicznych**, startupowe inkubatory, czy inicjatywy w stylu **Google X (Moonshot Factory)**, gdzie prototypuje się szalone pomysły, by sprawdzić ich realność. Inny analog to programy w rodzaju **NASA Flight Opportunities** – tam oferuje się szybkie loty testowe dla prototypów technologii kosmicznych, aby zwiększać ich TRL (Technology Readiness Level) tanim kosztem ³². Prototypownia danych pełni taką rolę „podnoszenia TRL” pomysłów data science: od TRL1 (koncept teoretyczny) do TRL4-5 (działający prototyp w warunkach zbliżonych do rzeczywistych) ³³.

Podsumowując, prototypownia danych jest **habitem o szybkim cyklu życia projektów**, nastawionym na eksplorację innowacji i zminimalizowanie kosztu i czasu potrzebnych do oceny ich wartości.

Źródła wartości ekonomicznej

Mimo że prototypy same w sobie nie generują od razu przychodów, prototypownia danych tworzy wartość ekonomiczną w kilku ważnych aspektach:

- **Redukcja ryzyka inwestycyjnego (de-risking):** Główną wartością jest *oszczędzenie potencjalnie ogromnych kosztów* wdrożenia wadliwego lub nieopłacalnego rozwiązania. Dzięki szybkiemu prototypowaniu, firma czy instytucja może wcześniej zdecydować, czy warto dany pomysł rozwijać. To jest jak wykonanie taniego zwiadu przed zainwestowaniem w dużą ekspedycję. W kategoriach finansowych, prototypownia poprawia **ROI projektów R&D**, bo odsiewa wcześniej pomysły o niskim ROI. Przykładem: jeśli zespół w 4 tygodnie stwierdzi, że model AI nie osiągnie wymaganej dokładności na dostępnych danych, można uniknąć wydania milionów na pełny system, który by potem zawiódł. To pewna *wartość opcji* – płacimy małą premię za informację, czy realizować dużą inwestycję lub z niej zrezygnować.

- **Inkubacja nowych produktów i przewagi konkurencyjnych:** Udane prototypy mogą przeistoczyć się w **nowe produkty lub usprawnienia** istniejących usług, co przynosi bezpośrednie przychody lub oszczędności. Prototypownia jest jak wydział *rozwijania IP* – generuje patenty, know-how, które następnie mogą być komercjalizowane. Nawet nieudane prototypy często generują spin-offy: np. technika opracowana przy okazji prototypu A może znaleźć zastosowanie w innym projekcie B. W ten sposób buduje się **portfel technologii**, z którego czerpią inne jednostki.

- **Zwiększenie efektywności R&D (skraca time-to-market):** Habitat specjalizuje się w prototypowaniu, więc wypracowuje metodologie, narzędzia i biblioteki przyspieszające eksperymenty. Tym samym koszt i czas typowego cyklu R&D spada. Ekonomicznie oznacza to, że **za tę samą kwotę** można zbadać więcej pomysłów albo że firma pierwsza wdroży nowe rozwiązanie (co daje przewagę zdobycia rynku). Według analiz, skrócenie cyklu R&D np. o 6 miesięcy w branży o szybkim postępie może być warte setki milionów, bo pozwala wcześniej generować przychody i uniknąć utraty udziału rynku.

- **Dostarczanie benchmarking i danych testowych:** Prototypownia tworzy również **zbiory danych i protokoły testowe**, które mogą mieć wartość same w sobie. Np. projekt prototypu może wymagać zebrania nowego rodzaju danych (choćby nagrania z dronów w jakimś formacie). Te dane, po zakończeniu prototypu, mogą stać się cennym zbiorem do licencjonowania lub do wykorzystania w kolejnych projektach. Również wypracowane *benchmarki* – powiedzmy procedura testowania wydajności modelu w warunkach zbliżonych do rzeczywistości – mogą być zaoferowane partnerom czy standaryzowane w branży. - **Edukacja i rozwój kompetencji:** Uczestnictwo w intensywnych prototypach **podnosi kwalifikacje zespołu**. Każdy prototyp to praktyczna lekcja: zespół uczy się nowych bibliotek, nowych metod, nowej domeny zastosowania. To buduje kapitał ludzki, który jest niezwykle cenny. Wartość tego przejawia się np. w wyższej produktywności pracowników w przyszłych projektach, czy w reputacji – eksperci z prototypowni mogą stać się uznanymi liderami w branży, co przyciąga kolejnych utalentowanych ludzi i projekty.

Choć prototypownia może nie generować bezpośredniego „przychodu ze sprzedaży” (jeśli działa jako jednostka wewnętrzna firmy), jej ekonomiczny sens manifestuje się w **poprawie wskaźników finansowych całej organizacji**: wyższa **Stopa Zwrotu z Innowacji (ROI z portfela projektów)**, wyższa **wartość obecna netto** dzięki redukcji ryzyk i szybszemu zwrotowi, oraz w **wartości intelektualnej** (patenty, licencje). W niektórych przypadkach prototypownia może też działać usługowo dla innych – np. realizować prototypy na zlecenie (outsourcing R&D). Wtedy generuje przychód bezpośrednio – np. jako projekt badawczy finansowany przez klienta czy grant (por. programy typu **NASA SBIR**, gdzie firma dostaje środki na prototypowanie pomysłu, a w zamian NASA potencjalnie uzyskuje prawo pierwszeństwa do wyników). W ramach SBIR małe firmy za Phase I dostają ~150-300k\$ na 6 miesięcy, a Phase II ~1M\$ na 24 miesiące ¹⁶ – to de facto zapłata za prototypy, która podtrzymuje ekosystem innowacji. Zatem, prototypownia może ściągać takie fundusze, wpisując się w łańcuch innowacji finansowany ze źródeł zewnętrznych.

Rachunek opłacalności (CAPEX, OPEX, ryzyka, punkty równowagi)

Nakłady CAPEX prototypowni danych są stosunkowo niskie w porównaniu z innymi habitatami, bo idea to *“kupować tylko to, co niezbędne do testów”*. Zwykle wystarczy podstawowa infrastruktura: kilka

uniwersalnych serwerów, stacje robocze dla zespołu, oprogramowanie analityczne (często open source lub licencje edukacyjne), przestrzeń biurowa/laboratoryjna. Jeśli potrzebny jest specjalistyczny sprzęt do prototypu, raczej się go **wypożycza** lub korzysta z partnerstw. Przykładowo, do prototypu analizy obrazów z nowego sensora LIDAR, zamiast kupować drogi sensor, habitat dogaduje się z dostawcą, by użyć egzemplarza testowego lub żeby prototyp przeprowadzić w siedzibie dostawcy. Zdarza się też, że prototypownia wykorzystuje **poligony testowe** cudzej infrastruktury – np. program **Flight Opportunities** pozwala badaczom lecieć ze swoim prototypowym sprzętem na rakiecie suborbitalnej, zamiast samemu budować raketę ³². CAPEX idzie więc głównie na *uniwersalne zasoby*, a specyficzne – na zasadzie „lease or borrow”.

Koszty operacyjne (OPEX) obejmują wynagrodzenia załogi (która jest wysoce wykwalifikowana, więc to największa pozycja), koszty zakupu danych (czasem trzeba zapłacić za dostęp do pewnych zbiorów do testów prototypu), opłaty za usługi chmurowe używane w eksperymentach, oraz koszty materiałów i energii. Te koszty są dość elastyczne – rosną kiedy jest dużo równoległych prototypów w toku, maleją gdy jest przerwa. Ponieważ prototypy są krótkie, budżet jest często alokowany per projekt. Wiele prototypowni stosuje mechanizm **wewnętrznego funduszu inwestycyjnego**: alokuje z góry X budżetu rocznie i rozdziela go na projekty. Jeśli projekt chce iść dalej (phase II), musi uzasadnić dofinansowanie. To sprawia, że OPEX jest mocno powiązany z portfelem – nie ma stałej produkcji, raczej *serie eksperymentów*.

Ryzyka finansowe prototypowni są dość niskie na poziomie pojedynczego projektu (bo z definicji ograniczamy nakłady). Największe ryzyko to **systematyczne niepowodzenia** – jeśli zbyt długo żaden prototyp nie przełoży się na korzyść, kierownictwo może zakwestionować sens istnienia takiej jednostki. Jest to coś w rodzaju ryzyka portfela: *czy wśród wielu strzałów pojawi się wystarczająco udany*, by pokryć koszty pozostałych. To analogia do funduszu venture capital: większość startupów padnie, ale ważne by jedna „rakietą” odniosła sukces 100x – w przypadku prototypowni „sukcesem” może być jedno rozwiązanie wdrożone, które przyniesie ogromne zyski czy oszczędności. Np. jeśli w ciągu 3 lat prototypownia wypuści prototyp technologii, który staje się nowym produktem o przychodach rzędu dziesiątek milionów, to opłaci to wszystkie drobne eksperymenty.

Punkt równowagi bywa trudny do formalnego wyznaczenia – bo to działalność o charakterze inwestycyjnym. Można liczyć w kategoriach *Real Options*: każda wydana kwota na prototyp to wykupienie opcji – zyski z niej zrealizują się, jeśli prototyp posłuży do wdrożenia. W praktyce firmy oceniają to *ex-post* – np. po 5 latach istnienia labu R&D patrzą, ile nowych produktów czy usprawnień powstało dzięki niemu i jaki generują cashflow. Jeżeli te korzyści > koszty labu, to break-even został osiągnięty. Często tak się dzieje, bo jeden kluczowy sukces pokrywa liczne drobne porażki.

Dodatkowo prototypownia korzysta z zewnętrznego finansowania (granty, SBIR, dofinansowania państwowe, partnerstwa). To zmniejsza obciążenie własne. Np. startup budujący prototyp technologii kosmicznej może dostać do 1,5 mln \$ w SBIR Phase II ¹⁶ – dla małej prototypowni to może pokryć większość rocznego budżetu. Tego typu fundusze można traktować jako *przychód* jednostki, co przybliża równowagę.

Ryzyka niefinansowe: możliwość wycieków informacji (przez luźne, eksperymentalne podejście do bezpieczeństwa, bo prototypy mogą nie mieć wszystkich zabezpieczeń – a to ryzyko reputacji), ryzyko demoralizacji zespołu przy serii porażek (co może skutkować utratą talentów), oraz ryzyko braku akceptacji wewnątrz firmy (działy operacyjne mogą ignorować wyniki prototypów z efektem „not invented here”).

Generalnie, prototypownia ma sens opłacalnościowy w środowiskach o wysokiej niepewności i wartości innowacji. W takich warunkach mały CAPEX i kontrolowany OPEX plus duże potencjalne outputy dają

bardzo korzystne **wartości oczekiwane**. Z literatury venture i R&D wynika, że parę-kilkanaście procent projektów R&D generuje 1/10 część całego zwrotu, reszta jest inwestycją w wiedzę ³⁴ ³⁵ (np. w SBIR 0,5% projektów dało >\$50M sprzedaży, top10 projektów ~30% całości sprzedaży programu ³⁴ ³⁶). Wiedząc o tym, prototypownia powinna tak planować portfel, by mieć dość „strzałów na bramkę” i liczyć, że kilka trafi do siatki.

Mechanizmy redukcji kosztów i maksymalizacji wartości

Prototypownia danych wdraża szereg mechanizmów, aby **pracować możliwie oszczędnie, a jednocześnie wyciskać maksimum wiedzy z każdego eksperymentu**:

- **Zwinne eksperymentowanie i minimalny prototyp (MVP):** Podstawową zasadą jest tworzenie **MVP (Minimum Viable Prototype)** – prototypu minimalnej koniecznej złożoności, by odpowiedzieć na pytanie badawcze. Unika się budowania zbędnych funkcjonalności. Dzięki temu koszty i czas wykonania są minimalne. Jeśli np. testujemy nowy algorytm, nie tworzymy od razu pełnej aplikacji z UI – wystarczy skrypt i kilka wykresów dla weryfikacji koncepcji. Mechanizm ten redukuje *przeinwestowanie w prototyp*, co często bywa bolączką klasycznych projektów (gdzie buduje się zbyt rozbudowany system zanim pozna się jego wartość).
- **Wykorzystanie istniejących narzędzi i open-source:** Prototypownia maksymalnie korzysta z **dostępnych bibliotek i platform** zamiast pisać wszystko od zera. Istnieją setki open-source'owych narzędzi ML, symulatorów, repozytoriów modeli (Hugging Face, scikit-learn, etc.). Zamiast wynajdywać koło na nowo, eksperyment integruje gotowe komponenty. Również **dane** czerpie się z publicznych zbiorów (jeśli to możliwe), np. do prototypu rozpoznawania obrazów medycznych używamy open dataset, a dopiero jak zadziała, myśli się o danych własnych. To dramatycznie obniża koszty – licencje open-source są darmowe, a czas pracy oszczędzony.
- **Symulacje i cyfrowe bliźniaki:** Zamiast budować kosztowne fizyczne prototypy czy testować w realu, habitat korzysta z **symulacji komputerowych** i modeli *digital twin*. Np. zanim przetestujemy algorytm sterowania flotą robotów w magazynie rzeczywistym (co wymagałoby fizycznych robotów), testujemy go w symulatorze 3D z modelami robotów. **Digital twin** potrafi wiernie odzwierciedlać warunki (są cyfrowe repliki np. procesów przemysłowych). Pozwala to zebrać dane testowe dużo taniej i bezpieczniej. ESA w projekcie MELiSSA (zamknięty ekosystem) też najpierw robiła modele symulacyjne poszczególnych bioreaktorów, zanim scala je w pilot plant ³⁷. W prototypowni taka praktyka zmniejsza koszty materiałowe i przyspiesza iteracje (symulację można powtarzać setki razy, co w rzeczywistości byłoby nieosiągalne).
- **Modułowe podejście i ponowne użycie komponentów:** Projekty prototypowe są różne, ale często mają wspólne elementy (np. moduł do wizualizacji danych, skrypt do trenowania sieci, pipeline ETL). Habitat buduje **bibliotekę wewnętrznych modułów**, które mogą być szybko adaptowane do nowych prototypów. To jak klocki Lego – nowy prototyp = nowe ułożenie klocków, a nie budowa od zera. W skali organizacji to *reuse*, co oszczędza koszty i zapewnia pewną jakość (sprawdzony komponent jest mniej zawodny).
- **Crowdsourcing i otwarta współpraca:** Aby szybko znaleźć rozwiązania lub zebrać pomysły, prototypownia może korzystać z **crowdsourcingu**. Np. organizuje mini-konkurs (jak Kaggle) – udostępnia anonimowe dane i problem społeczności, by zebrać rozwiązania prototypowe z zewnątrz. Albo współpracuje open-source'owo – publikuje część kodu i przyjmuje wkład od wolontariuszy. To mechanizm "większego mózgu", który niskim kosztem (nagrody, uznanie) może przynieść prototyp lub jego fragment. W idei stigmergii – upublicznienie problemu jest jak

zostawienie *feromonu*, który przyciąga zainteresowanych do wspólnej pracy nad prototypem, co bywa skuteczne w trudnych problemach.

- **Stopniowe zwiększanie realizmu (spiralą TRL):** Mechanizm planowania prototypów jest często **wieloletapowy**. Najpierw testujemy w warunkach bardzo uproszczonych (TRL 3-4), jeśli się uda – idziemy poziom wyżej: prototyp w środowisku symulowanym z realizmem (TRL 5-6). Potem ewentualnie demonstracja w małej skali realnej (TRL 7). Taka *spiralą* pozwala stopniowo wkładać więcej zasobów tylko w pomysły, które przechodzą kolejne bramki. NASA i ESA stosują TRL-y właśnie by decydować, kiedy technologia jest gotowa na drogi pełnoskalowy test ³⁸ ³⁹ . Prototypownia trzyma się tego: nie pakujemy od razu prototypu do pełnej produkcji – najpierw mini-demonstrator. To minimalizuje koszty porażki: jeśli prototyp nie daje rady na poziomie TRL5, rezygnujemy, zanim zrobimy kosztowny pilot TRL7.
- **Uczenie się z prototypów (metaanaliza):** Habitat nie tylko wykonuje prototypy, ale też je **analizuje przekrojowo**. Prowadzi *post-mortem* każdego projektu: co zadziało, co nie, czemu. Te lekcje zbiera i karmi nimi kolejne prototypy. W ten sposób następuje *metauczenie* organizacji – staje się coraz lepsza w przewidywaniu, jak podejść do kolejnych innowacji. Wartością jest tu zmniejszenie błędów powtarzalnych i szybsze dochodzenie do sedna. To tak jak systematyczny **kaizen** w przemyśle – ciągłe doskonalenie procesu prototypowania. Efekt: prototypy stają się coraz tańsze i celniejsze.

Rola człowieka w pętli (HITL)

W prototypowni danych **ludzie odgrywają rolę kluczową na każdym etapie** – to środowisko silnie zorientowane na kreatywność i wiedzę ekspercką, więc rola AI jest tu raczej narzędziowa. Mimo korzystania z automatyzacji, prototypowaniem kieruje ludzka ciekawość i intuicja:

- **Faza generowania pomysłów:** Człowiek (zespół) identyfikuje problem wart rozwiązania i proponuje koncept AI, który mógłby to poprawić. Tutaj ludzka kreatywność i znajomość kontekstu dominują. Mogą wesprzeć brainstorming narzędziami (np. AI do podpowiedzi literaturowych), ale *inicjatywa jest ludzka*. Eksperci dziedzinowi formułują pytania badawcze: np. "Czy za pomocą AI da się przewidzieć awarię maszyny 24h wcześniej?". Ludzki *instynkt i doświadczenie* decydują, czy prototyp jest w ogóle sensowny do spróbowania.

- **Projektowanie eksperymentu:** To ludzie (data scientiści) wybierają metodologię: np. planują, że użyjemy sieci LSTM, bo to intuicyjnie pasuje do sekwencji danych. Określają, jakie dane zebrać, jakie miary sukcesu. Oczywiście korzystają z wiedzy (i tu np. AI asystent kodowania może im pomóc pisać szybciej kod, albo chatGPT podsunąć pomysł w trakcie programowania), ale człowiek podejmuje decyzje architektoniczne.

- **Realizacja prototypu – człowiek w pętli testowania:** Podczas budowania i trenowania modeli, ludzie **interaktywnie pętla się z wynikami**. Prototypownia to typowe miejsce stosowania **metod human-in-the-loop jak Active Learning** – analityk patrzy na wyniki częściowe, decyduje, które przykłady doanotować, poprawia parametry, uruchamia ponownie. Człowiek tu stale ocenia: np. generuje próbne wyjścia modelu i *na oko* ocenia sensowność. W odróżnieniu od trybu produkcyjnego, tu to nie jest pipeline w pełni automatyczny – raczej *dialog człowieka z modelem*: "co jeśli zmienię tę hiperparametr? a co jeśli dodam tę cechę?". Dzięki temu prototyp jest optymalizowany inteligentnie (bo człowiek kieruje poszukiwaniem w przestrzeni możliwości).

- **Ocena i interpretacja wyników:** Gdy prototyp da wynik, człowiek (a najlepiej zespół mieszany: inżynier + ekspert domenowy) **interpretuje** go: czy to dobrze czy źle, co to oznacza. Człowiek tu wnosi kontekst, którego AI nie ma. Np. model osiąga 70% trafności – algorytm mówi "ok", ale inżynier wie, że dotychczasowe rozwiązanie miało 90%, więc 70% jest za słabo (nawet jeśli np. przekroczyło inny baseline). Albo odwrotnie: model tylko 60% trafny, ale ekspert domenowy mówi "w tej dziedzinie to i tak rewolucja, bo ludzie mieli 55%". Bez człowieka taka ocena byłaby niemożliwa. Dodatkowo, ludzie

identyfikują *przyczyny* – np. patrzą na błędne przypadki i odkrywają, że model zawsze myli się w pewnej podkategorii, co może sugerować jak go poprawić.

- **Decyzja i dalsze działania:** Ostatecznie to człowiek decyduje, co zrobić z prototypem: czy "zabić" projekt, czy iterować dalej, czy przekazać do działu wdrożeń. Decyzje tego typu nie są czysto liczbowe, często wymagają *intuicji i wyczucia strategicznego*. Ludzie biorą tu pod uwagę nie tylko metryki modelu, ale np. potencjał rynkowy, alignment z celami firmy, zasoby potrzebne do wdrożenia. To jak panel oceniający startupy – prototypownia może mieć radę (ludzi) oceniającą wyniki i decydującą o dalszych krokach, analogicznie jak w SBIR panel ekspercki decyduje, które projekty Phase I warte są Phase II 40

41 .

Krótko mówiąc, w prototypowni **człowiek jest wszechobecny w pętli** – jest inicjatorem, eksperymentatorem, kontrolerem jakości i decydentem. AI jest tu bardziej "asystentem laboranta": przyspiesza obliczenia, automatyzuje nudne kawałki (np. przetwarzanie tysiąca plików), może wspierać analizę (np. klastrowanie danych by człowiek zobaczył wzorce). Ale nie zastępuje człowieka, bo faza prototypu wymaga *wiele twórczego myślenia, adaptacji i interpretacji*, co jest atutem Homo Sapiens.

Ta intensywna rola HITL jest powodem, dla którego prototypownie muszą zatrudniać utalentowanych, wszechstronnych ludzi i dbać o ich rozwój. Jednocześnie pracując nad prototypami ludzie **zdobywają unikalną wiedzę** (np. o tym co nie działa i czemu), co jest bezcenne. W tym sensie, nawet nieudany prototyp zwiększa *kompetencje ludzi*, co potem przenosi się na sukces kolejnych (wiedzą już czego unikać, albo nową technikę opanowali). Zatem synergia: im więcej ludzie są "w pętli" prototypowania, tym stają się lepszymi prototypistami.

Przepływ danych: od surowych danych do produktu eksperymentalnego

W prototypowni przepływ danych jest bardziej ad-hoc i elastyczny niż w trybie produkcyjnym – bo każdy eksperyment może wymagać innego podejścia. Mimo to pewien generyczny schemat można zarysować:

1. **Pozyskanie/Generacja danych surowych:** Dla każdego prototypu trzeba zidentyfikować, skąd weźmiemy dane do analizy. Czasem organizacja już ma dane historyczne (np. logi, bazy), wtedy to punkt wyjścia. Często jednak prototyp wymaga **akwizycji nowych danych**: np. przeprowadzenia krótkiego pomiaru, zrobienia ankiety, crawlingu internetu, skorzystania z open data. Jeśli danych jest mało lub brak, prototypownia generuje *dane syntetyczne*: np. symuluje czujnik lub augmentuje drobną próbkę (dodając szum, transformacje), by trenować model. Dane surowe w tym kontekście mogą być dość małe (w prototypie nie potrzeba milionów przykładów – tyle, by sprawdzić koncept). Np. do prototypu detekcji defektów można zacząć od 100 zdjęć wybrakowanych produktów, nawet jeśli docelowo potrzeba by 100k. Tak więc surowe dane w prototypowni są często *podzbiorem docelowych danych*, ale starannie wybranym/przygotowanym.
2. **Anotacja i przygotowanie danych:** Otrzymane dane rzadko są od razu użyteczne. Trzeba je oczyścić, oznakować cechami, ewentualnie wzbogacić. Tutaj duży nacisk na **szybką, ręczną anotację**: np. zespół własnoręcznie oznacza etykiety w kilkuset próbkach (nie opłaca się tworzyć wielkiej pipeline – lepiej ręcznie, bo to prototyp). Wspomagają się prostymi narzędziami (Labeling tool). Tworzą też *zbiór testowy* – bo kluczowa jest walidacja. W prototypach często *dzielimy dane w inny sposób niż w produkcji*: np. nie silimy się na rozbudowaną walidację krzyżową – prościej, np. 80/20. Tworzy się też **metadane eksperymentu**: notatki co która konfiguracja oznacza, jakie parametry testujemy – to buduje dokumentację do analizy wyników.
3. **Budowa modelu prototypowego:** Na przygotowanych danych wykonuje się **szybkie iteracje modelowania**. To może być trening ML, test różnych algorytmów, doboru hiperparametrów – ogółem, generowanie *kandydatów rozwiązania*. W prototypowni liczy się szybkość – być może

używa się losowego przeszukiwania hiperparametrów, bo grid search byłby za wolny. Można testuje się 10 różnych algorytmów z minimalnym strojeniem (a nie optymalizuje jednego do upadłego). W każdej iteracji model generuje wyniki (predykcje, klasyfikacje), które *ludzie oglądają* – np. rysują wykres, patrzą na confusion matrix, czy sprawdzają kilka predykcji ręcznie. Ten proces to tak naprawdę **pętla eksperymentalna**: (a) wybierz konfigurację -> (b) uruchom -> (c) oceń wyniki -> (d) zdecyduj co zmienić i powtórz. Pętla trwa aż do osiągnięcia albo zadowalającego wyniku, albo wyczerpania sensownych opcji (co sugeruje granicę możliwości).

4. **Weryfikacja i walidacja prototypu**: Gdy zespół uzna, że ma działający prototyp (np. model o pewnej skuteczności lub pipeline analityczny generujący jakiś output), następuje **test w warunkach docelowego użycia** – na tyle, na ile to możliwe. To jest kluczowe: sprawdzić, czy prototyp nie jest „oszukany” pod warunki labowe. Na przykład, jeśli to model do predykcji awarii, testuje się go na całkiem nowym zbiorze dni z historii, aby zobaczyć czy faktycznie alarmował przed prawdziwymi awariami (to wstępny proof-of-value). Często walidacja prototypu obejmuje też **ocenę jakości przez ekspertów**: ludzie poza zespołem prototypu patrzą na wyniki. Np. lekarz ogląda kilkanaście zdjęć z zaznaczonymi przez AI zmianami i ocenia, czy faktycznie tam są zmiany – bo statystyki to jedno, ale ekspert może wyłapać, że AI regularnie zaznacza też artefakty, co statystycznie ginie w liczbach, ale klinicznie jest problemem. Zatem następuje *peer review* prototypu.

5. **Opracowanie wyników i wnioski**: Z surowych wyników testów zespół **opracowuje „produkt prototypowy”** – to zwykle raport lub prezentacja + ewentualnie demonstracja działającego modelu. Ten produkt to *wiedza*: np. „model X osiągnął Y% skuteczności, co jest o Z% lepsze od obecnej metody, jednak wymaga danych takiej a takiej jakości”. Albo: „Pomysł okazał się niewypałem, bo sygnał w danych jest zbyt słaby; zalecamy nie kontynuować, bo to ograniczenie fizyczne.” Produktem prototypowni jest więc **zestaw rekomendacji i dowodów**. Jeśli prototyp jest udany, do raportu dołącza się często *plan dalszych kroków* (np. *co potrzeba by wdrożyć*). Jeśli nieudany – *analizę przyczyn* (to cenna informacja, by nie powtarzać).

6. **Decyzja i archiwizacja**: Jak wspomniano, ludzie decydują co dalej. Jeżeli decyzja to zakończyć, prototyp zostaje **zarchiwizowany**: kod, dane i wyniki są zachowane w repozytorium (by w przyszłości można było do nich sięgnąć przy podobnym pomysle lub do uczenia). Jeżeli decyzja to kontynuować (np. kolejna faza lub przekazanie do działu wdrożeń), to *przepływ danych przechodzi do innego trybu*: prototypownia może przekazać pałeczkę innemu zespołowi, udostępniając mu wszystkie dane i modele. W idealnym scenariuszu następuje też feedback: po wdrożeniu, wyniki z prawdziwej operacji wracają informacyjnie do prototypowni („nasz prototyp w skali faktycznie działa/działa gorzej niż myśleliśmy bo...”), co karmi kolejne projekty.

W odróżnieniu od stałego pipeline produkcyjnego, przepływ danych w prototypowni jest **jednorazowym cyklem** (choć pętle wewnątrz powtarzają się iteracyjnie). Trzeba jednak zadbać, by finalne dane (wyniki, odkrycia) nie zginęły – stąd archiwizacja i raportowanie jest kluczowe. Protokoły jak **FAIR data** są przydatne: wyniki powinny być *Findable* (*łatwo odszukiwalne w przyszłości*) i *Reusable* (*opisane tak, by dały się wykorzystać przez kogoś innego*) ⁹. Prototypownia zazwyczaj buduje wewn. wiki lub repozytorium wiedzy z projektów.

Krytyczne heurystyki sukcesu

Aby prototypownia danych skutecznie spełniała swoją rolę i generowała wartość, stosuje następujące heurystyki zarządcze:

- **Strzelaj celnie, ale tanio (frugal innovation):** Priorytetem jest osiągnięcie *wystarczająco dobrych* rezultatów za minimalny koszt. Heurystyka brzmi: „**czy to jest najprostszy eksperyment, który może zadziałać?**” – jeśli nie, upraszczaj. Zespół musi ciągle pilnować, by nie przekombinować prototypu. Często przypomina się zasadę **80/20**: 20% wysiłku może dać 80% wglądu. W prototypowni nie dążymy do perfekcji, a do intuicji czy coś jest obiecujące. To zapobiega *gold-platingowi* (przeinwestowaniu prototypu).
- **Ucz się z porażek, aby były wartościowe:** Jak wspomniano, nie każdy prototyp da sukces, ale heurystyka to „**żadna porażka nie jest na darmo**”. Formalizuje się to poprzez retrospektywy, spisywanie wniosków i dzielenie się nimi. Jeśli np. prototyp wskazał, że pewien sygnał w danych nie koreluje z wynikiem, to wiedza ta jest rzadkością – powinna być zapamiętana. Może ona wskazać organizacji by nie marnować więcej zasobów w tym kierunku. W pewnym sensie prototypownia stara się *monetyzować porażki wewnętrznie jako wiedzę*, co jest przeciwieństwem kultury, gdzie porażki się tuszuje.
- **Zarządzaj portfelem eksperymentów (diversification):** Aby zwiększyć szanse, habitat stosuje heurystykę **dywersyfikacji pomysłów** – prototypuje rzeczy o różnym profilu ryzyka. Nie wszystkie projekty mogą być „moonshotami”; w miksie są też *low-hanging fruits* (łatwe usprawnienia). To gwarantuje pewne „wygrane”, które budują morale i uzasadniają istnienie labu, podczas gdy ryzykowne projekty mogą cziąć duży payout. Podobnie jak w portfolio inwestycyjnym: trochę obligacji, trochę akcji. W R&D literaturze to zalecane podejście do innowacji – zrównoważenie exploit (ulepszaj obecne) i explore (nowe przełomowe).
- **Utrzymuj bliski kontakt z „realem”:** Prototypy muszą być weryfikowane realiami, inaczej grozi pułapka „*cud w laboratorium, fiasko w praktyce*”. Heurystyka to „**jak najszybciej testuj z realnymi użytkownikami/warunkami**”. Np. zbudowałeś prototyp aplikacji – daj ją choćby kilku rzeczywistym użytkownikom do spróbowania, niech powiedzą co nie gra. Zrobiłeś model predykcyjny – pokaż jego wyniki fachowcowi operacyjnemu, czy mają sens. To chroni przed zbyt akademickim podejściem i tchnięciem w prototyp fałszywego poczucia sukcesu. Często drobny test realny ujawnia duże problemy (np. model działa w labie na czystych danych, a w praktyce dane są hałaśliwe i wyniki spadają). Lepiej dowiedzieć się tego wcześniej (gdy pivot jeszcze możliwy).
- **Zabezpiecz własność intelektualną wcześniej:** Jeśli prototyp wskazuje na coś bardzo obiecującego, heurystyka to „**patentuj/nadawaj prawa zanim upublicznisz**”. Prototypownia może generować pomysły, które są nowatorskie – by nie stracić do nich praw, organizacja powinna szybko ocenić, czy warto np. złożyć zgłoszenie patentowe czy zachować tajemnicę handlową. Zwłaszcza że prototypy często muszą konsultować się z zewnętrznymi partnerami lub wykorzystywać cudze komponenty. Trzeba dbać, by kluczowe elementy IP pozostały wewnątrz. W NASA SBIR np. małe firmy pilnują IP, bo wiedzą że owoce prototypu to ich kapitał. W firmie komercyjnej prototypownia w porozumieniu z działem prawnym powinna mieć check-list: nowy algorytm – patent; nowy unikalny zbiór danych – decyzja czy nie upublicznić go w całości. To z kolei maksymalizuje wartość – bo jak prototyp przejdzie dalej, firma ma już ochronę prawną i może czerpać wyłączną korzyść.

- **Zachowaj kadre głodną wyzwań:** Sukces prototypowni to *ludzie z pasją do innowacji*. Heurystyka miękka: „**utrzymaj kulturę ciekawości i tolerancji błędów**”. Lider labu powinien doceniać nawet te zespoły, którym nie wyszło, jeśli pracowały metodycznie i wyciągnęły wnioski. Chwalić publikacje czy patenty, budować reputację labu. Dbać, by utalentowani się nie nudzili (dawać im coraz to nowe ambitne prototypy). Rotować ludzi między projektami by uczyli się nowych rzeczy. To heurystyka bardziej HR, ale krytyczna – bo prototypownia opiera się na kreatywności i motywacji ludzi. Bez tego nawet najlepsze procesy nie zadziałają.

Stosując te zasady, prototypownia może utrzymać wysoką skuteczność pomimo ryzykownego charakteru swojej działalności. Pozwala to generować strumień innowacji z akceptowalnym poziomem kosztów i wielkim potencjałem upsidu – co czyni ten model nieodzownym elementem organizacji nastawionych na rozwój w niepewnym, dynamicznym otoczeniu.

(Dalsza część raportu, obejmująca ekosystem 4: Habitat regeneracyjny oraz ekosystem 5: Środowisko certyfikacji i benchmarków, wraz z tabelarycznym porównaniem modeli i podsumowaniem, znajduje się w kolejnych sekcjach dokumentu.)

Porównanie modeli – tabela

Poniżej przedstawiono zestawienie kluczowych cech pięciu przeanalizowanych ekosystemów Human-AI w formie tabelarycznej:

Aspekt	Laboratorium Społecznej SI (Social-AI Lab)	AI-ops jako usługa (AI-ops-aaS)	Prototypownia danych (Data Lab)	Habitat regeneracyjny (Closed-loop AI Habitat)	Środowisko certyfikacji (Benchmark Cert Lab)
Cel nadrzędny	Inkluzywne współtworzenie rozwiązań AI z społecznością.	Automatyczne świadczenie usług AI dla wielu użytkowników (skala, dostępność 24/7).	Szybkie testowanie innowacji i pomysłów data science (de-risking R&D).	Samowystarczalna produkcja danych i decyzji w zamkniętym ekosystemie (sustainability).	Niezależna weryfikacja, ocena i certyfikacja systemów AI (bezpieczeństwo standardy).
Przykłady/Analogie	Sociaal AI Lab Rotterdam (miasto + mieszkańcy) ² ; Living labs miejskie.	OpenAI API (model GPT jako usługa) ¹⁸ ; Chmurowe platformy MLaaS.	Google X, inkubatory NASA (Flight Opportunities) ³² ; programy SBIR Phase I/II ¹⁶ .	ISS National Lab + MELiSSA (zamknięta biosfera) ⁴² ⁴³ ; bazy analogowe (EBIOS) ⁴³ .	Underwriters Labs dla AI; t autopilotów (Euro NCAP c aut. pojazdów MLPerf (konsorcjum benchmarków

Aspekt	Laboratorium Społecznej SI (Social-AI Lab)	AI-ops jako usługa (AI-ops-aaS)	Prototypownia danych (Data Lab)	Habitat regeneracyjny (Closed-loop AI Habitat)	Środowisko certyfikacji (Benchmark Cert Lab)
Źródła wartości ekonomicznej	Lepsze dopasowanie technologii do potrzeb (uniknięcie nietrafionych wdrożeń); budowanie kapitału społecznego i zaufania (przyspiesza adopcję e-usług). Wartość pośrednia: insighty społeczne, edukacja (mieszkańcy stają się kompetentni cyfrowo).	Bezpośrednia monetyzacja: opłaty za zapytania, subskrypcje API (np. \$700k koszt/dzień – finansowany przez użytkowników) ²² . Wartość rośnie z skalą: im więcej klientów, tym więcej danych do ulepszania usług (efekt sieciowy). Nowe modele licencjonowania – np. udział w zyskach klientów (royalty) ¹⁴ .	Oszczędność przez zapobieganie złym inwestycjom (każdy prototyp to „ubezpieczenie”); tworzenie IP (patenty, protokoły) do przyszłej komercjalizacji. Usprawnienie procesu innowacji (time-to-market krótszy, ROI R&D wyższe). Możliwe granty i dofinansowania (obniżające koszty własne).	Redukcja kosztów operacji przez autonomię (mniej interwencji) – np. ISS regenerując 50% wody oszczędza X\$ w dostawach. Unikalne dane z izolowanego środowiska (sprzedaż eksperymentów kosmicznych – potencjał \$110 mld market open, \$900 mln product revenue) ⁴⁴ . Możliwość turystyki edukacyjnej (jak EBIOS, \$3-6k za pobyt) ⁴⁵ .	Zapobieganie awariom i skandalom – oszczędność potencjalnych kar i strat (np. certyfikowane nie spowodują wypadku -> uniknięte koszty ogromne). Przychody z usług testów (firmy płacą za audyt/ certyfikację AI jak za TÜV CE). Tworzenie standardów branżowych kontrola nad rynkiem (wartość strategiczna)
CAPEX (infrastruktura)	Niski – fizyczna przestrzeń, komputery dla warsztatów, czasem mobilna pracownia. Wykorzystanie istniejących zasobów (biblioteki, szkoły).	Bardzo wysoki – centra danych (miliony \$ na GPU, serwery, chłodzenie). Np. > \$1 mld inwestycji w superkomputer dla OpenAI ²¹ . Partnerstwa strategiczne (Big Tech) jako źródło kapitału.	Niski/umiarkowany – gł. uniwersalne serwery, oprogramowanie open source. Specjalistyczny sprzęt wypożyczany do testów. Zasada: nie kupuj, jeśli możesz symulować/ użyć chmury.	Bardzo wysoki – budowa habitatów, modułów podtrzymania życia, sensoryka full redundancy. Np. ISS: >\$100 mld budowa; analog EBIOS planowany koszt niejawni (wielomilionowy).	Umiarkowane – laboratoria testowe, tzw. sandboxy regulacyjne (urządzenia testowe). Koszt bazy danych testowych, infrastruktury symulacji zagrożeń. Człowiek wykorzystanie istniejących zasobów, poligonów (mimo że testowy a

Aspekt	Laboratorium Społecznej SI (Social-AI Lab)	AI-ops jako usługa (AI-ops-aaS)	Prototypownia danych (Data Lab)	Habitat regeneracyjny (Closed-loop AI Habitat)	Środowisko certyfikacji (Benchmark Cert Lab)
OPEX (operacje)	<p>Stały koszt personelu (animatory, trenerzy, analitycy) i organizacji wydarzeń. Niewielkie koszty sprzętu/energii. Mogą być dotowane ze środków publicznych (trudne do uzasadnienia ROI krótkoterminowe).</p>	<p>Bardzo wysoki i zmienny – energia elektryczna (prąd dla GPU – ChatGPT ~\$700k/dzień²², chłodzenie (też zużycie wody⁴⁶), wynagrodzenia specjalistów AI (kilkaset osób). Skalowanie personelu wolniejsze niż skali zapytań (automatyzacja).</p>	<p>Głównie płace niewielkiego zespołu ekspertów (największy składnik). Koszty projektowe: zakupy danych, opłaty chmurowe za eksperymenty, ewentualnie honoraria konsultantów zewn. Zmienny – zależy ile prototypów równolegle. Możliwość pokrycia części z grantów.</p>	<p>Wysoki – stałe utrzymanie załogi (jeśli habitat z ludźmi) lub nadzoru, koszty energii (np. sztuczne oświetlenie upraw, recykling wody), konserwacja systemów. ISS ~ \$3-4 mld rocznie utrzymanie. W habitatcie docelowo automatyka zmniejsza OPEX, ale wymaga redundancji (zapasowe części, roboty).</p>	<p>Umiarkowane etaty ekspertów do testów/odsłuchów, koszty aktualizacji zestawów testowych i infrastruktury audytu bezpieczeństwa. Część kosztów przerzucana na klientów płacących za certyfikację.</p>

Aspekt	Laboratorium Społecznej SI (Social-AI Lab)	AI-ops jako usługa (AI-ops-aaS)	Prototypownia danych (Data Lab)	Habitat regeneracyjny (Closed-loop AI Habitat)	Środowisko certyfikacji (Benchmark Cert Lab)
Mechanizmy obniżania kosztów	Współpraca z lokalnymi partnerami (wymiana zasobów za wartość niematerialną – udział mieszkańców); wolontariat i staże (część pracy ludzkiej za darmo + edukacja); open source i otwarte dane (niższe koszty licencji); co-creation zmniejsza ryzyko oporu przy wdrożeniach (większa skuteczność projektów).	Automatyzacja + optymalizacja zużycia; modele dopasowywane do tańszych chipów ⁴⁷ ; autoskalowanie zasobów; <i>self-healing</i> (mniej awarii -> mniej drogich napraw). Własne algorytmy optymalizujące koszty (np. routing zadań do najtańszych centrów w danych godzinach). Duże wolumeny -> hurtowy zakup energii i sprzętu taniej.	Lean prototyping: wykorzystanie symulacji zamiast fizycznych testów (taniej o rząd wielkości); iteracje MVP (nie budujemy więcej niż trzeba); reuse kodu z poprzednich prototypów; szybkie „killowanie” projektów bez perspektyw (żeby nie ciągnęły zasobów). Fin. zewnętrzne (granty) na pokrycie kosztów ryzykownych eksperymentów.	Pełna recykulacja zasobów (energia słoneczna, recykling woda/ powietrze – mniej dostaw z Ziemi) ⁴² ; automatyzacja prac rutynowych (mniej załogi ludzkiej); synergia procesów (odpady jednego modułu stają się surowcem innego – analogia ekosystemu naturalnego). Stigmergia między agentami – optymalne sterowanie (np. cyfrowe feromony sygnalizują gdzie potrzeba chłodzenia bardziej – dynamiczne równoważenie).	Standaryzacja procedur testowych (masowy przerób wielu systemów – niższy koszt produkcji system); narzędzia automatycznego testowania (np. fuzzing, generatory przypadków testowych) – ludzie tylko nadzorują; współpraca branżowa – koszty podzielone między wieloma firmami (konsorcja tworzą benchmark zamiast każda firma osobno).

Aspekt	Laboratorium Społecznej SI (Social-AI Lab)	AI-ops jako usługa (AI-ops-aaS)	Prototypownia danych (Data Lab)	Habitat regeneracyjny (Closed-loop AI Habitat)	Środowisko certyfikacji (Benchmark Cert Lab)
Rola człowieka (HITL)	Centralna – mieszkańcy jako współtwórcy (dają dane, oceniają AI) ⁴⁸ ; eksperci dbają o etykę, poprawiają błędy AI na bieżąco (AI asystuje tylko). Człowiek inicjuje projekty (wskazuje problemy społeczne), moderuje algorytmy (np. Digiderius – chatbot prowadzony we współpracy z ludźmi) ¹³ .	Znaczna, lecz w tle – ludzie projektują system i poprawiają modele poprzez RLHF (np. trenerzy poprawiają odpowiedzi GPT, co znacznie podnosi jakość ¹¹). W operacjach bieżących człowiek reaguje przy wyjątkach (eskalacja trudnych przypadków do eksperta), monitoruje wskaźniki jakości. Człowiek jako decydent biznesowy – wybiera modele, polityki, strategię cenową.	Krytyczna – ludzie to pomysłodawcy i eksperymentatorzy; ręcznie przygotowują dane, wybierają metody, interpretują wyniki. HITL w pętli eksperymentu: testują prototyp iteracyjnie, oceniając skutki każdej zmiany. Wnioski i decyzje (kontynuować/ przerwać) są w całości ludzkie – to ich doświadczenie stanowi o sukcesie lub brak (AI tu narzędzie).	Kluczowa w nadzorze i interwencji krytycznej – ludzie ustalają reguły kontroli (kiedy system ma alarmować, że parametry wychodzą poza normę), a w razie awarii czy niepewności przejmują sterowanie. Człowiek jako beneficjent – to dla załogi tworzy się przyjazne warunki (HITL ocenia komfort, bezpieczeństwo – np. astronauta raportuje co systemy nie przewidziały). Ponadto, ludzie są częścią pętli danych (np. ogród w MELISSA pielęgnowany częściowo przez ludzi dostarcza danych o interakcji człowiek-roślina).	Niezastąpiony jako arbiter prawdy – ludzie definiują kryteria testów, interpretują wyniki (np. czy błąd jest krytyczny czy akceptowalny), wydają ostateczny werdykt „zaliczony/nie”. Także opracowują trudno formalizowane testy (np. etyczne eval panel oceniający decyzje AI). Często wykorzystuje grupy eksperckie niezależnych (podejście rozproszonej oceny dla obiektywizmu). Człowiek reaguje też na próby obejścia – adaptuje benchmarki, widzi że AI nauczyły się „grać pod testy”.

Dane – przepływ i produkty

Dane obywatelskie -> warsztatowe prototypy -> rekomendacje dla władz. Surowe dane: opinie mieszkańców, lokalne zbiory miejskie. Metadane: kontekst społeczny (np. metryki inkluzji). Walidacja: wspólna ocena z mieszkańcami (czy rozwiązanie działa dla nich). Produkt danych premium: raporty z konsultacji, prototypowe aplikacje miejskie (np. lokalny chatbot), zwiększona świadomość potrzeb (dane jakościowe). Dystrybucja: publikacje, otwarte dane (częściowo) – ale z dbałością o prywatność.

Ciągły strumień danych klientów -> system AI -> odpowiedzi/ usługi. Surowe: zapytania tekstowe, obrazy, itp. w ogromnej skali. Pipeline automatyczny z walidacją końcową (filtrowanie toksycznych outputów) ⁴⁹. Produkty premium: natychmiastowe odpowiedzi (np. ChatGPT), insights (raporty, alerty – jak Maxar Sentry) ²⁰. Sprzedaż poprzez API, SaaS, licencje. Dane zwrotne (logi) -> doskonalenie modeli (feedback loop).

Dane projektowe (małe zestawy testowe, syntetyczne) -> eksperymenty -> wiedza. Surowe: wybrane przykłady reprezentatywne problemu; ewent. symulacje generują pseudo-dane. Walidacja: wydzielenie części na test, cross-check z ekspertem. Produkt: prototyp modelu lub demonstrator + dokumentacja wyników i rekomendacje. Dystrybucja: wewnętrzny raport lub prezentacja decyzyjna; ewent. patent; kody źródłowe w repo (do reuse).

Dane czujników i użytkowania -> algorytmy zarządzające -> recyrkulacja informacji. Surowe: tysiące strumieni sensorów (temp., CO2, wilgotność, poziom odpadów etc.), logi systemów AI, parametry życiowe załogi. Pętla danych: system sterowania na bieżąco optymalizuje (np. więcej światła -> rośliny rosną szybciej -> więcej tlenu dla ludzi). Walidacja: wbudowane czujniki redundantne (sprawdzają się nawzajem), człowiek-kontroler. Produkt danych: raporty eksperymentów (np. uprawa w mikrogravitacji), wytrenowane modele do sterowania autonomicznego. Część danych otwarta naukowo (np. ISS publikuje surowe dane dla badaczy) ⁵⁰.

Dane testowe system AI kandydat -> wyniki porównawcze certyfikat/raport Surowe: standaryzowane zestawy (np. obrazki, scenariusze symulacyjne), często tajne i uniemożliwiające „uczenie się testu”; również dane od producenta (dokumentacja modelu). Przetwarzanie przebiegu testu automatycznie + oceny ekspertów (HAI przy ocenie zachowania). Wynik: metryki (accuracy, robustność, itp.) vs. progi akceptacji. Produkt: certyfikat (jesli spełnione wymogi) lub niezgodność. Dane wyjściowe (benchmark scores) mogą być publikowane (rankingi), co motywuje firmę do poprawy (utrzymanie rzadkości danych – np. tylko topowe modele osiągnęły określony wynik).

(Źródła: Wypełnione w tabeli odniesienia ² ¹⁸ etc. wskazują przykładowe fragmenty wykorzystane przy opisie.)

Podsumowanie

Przedstawione pięć ekosystemów typu Human-AI – od społecznych laboratoriów miejskich, przez komercyjne chmury AI, prototypownie R&D, autonomiczne habitaty, po środowiska certyfikacji – ilustruje szerokie spektrum sposobów, w jakie ludzie i sztuczna inteligencja mogą współistnieć i współpracować w ramach złożonych systemów. Każdy model pełni odmienną rolę i operuje na innych zasadach, jednak pewne wspólne wnioski wyłaniają się z analizy:

- **Human-in-the-loop pozostaje krytyczny** we wszystkich typach ekosystemów, choć przyjmuje różne formy. W ekosystemach społecznych i eksperymentalnych (Lab Społeczny, Prototypownia) człowiek jest inicjatorem i kreatorem wartości, a AI narzędziem. W systemach maszynowych na wielką skalę (AI-ops, Habitat regeneracyjny) człowiek pełni rolę nadzorca i strażnika jakości, zapewniając bezpieczeństwo i kierunek rozwoju. W środowiskach certyfikacji to autorytet ludzki finalnie decyduje o zaufaniu do AI. To pokazuje, że **kompetencje ludzkie – intuicja, etyka, zdrowy rozsądek – są kluczowe** dla maksymalizacji wartości i minimalizacji ryzyka związanego z AI.
- **Dane stanowią centralny zasób i „produkt”** wszystkich habitatów, lecz ich rola się różnicuje. W labach społecznych dane są nośnikiem głosu społeczności i budują zaufanie (spełniając zasady FAIR, by mogły służyć szeroko). W AI-ops dane (w ogromnej ilości) są paliwem dla algorytmów i jednocześnie produktem (informacje na żądanie). W prototypowniach – dane są kosztem, który należy minimalizować, a ich najważniejszą funkcją jest generowanie *informacji decyzyjnej* (czy pomysł działa). W habitatcie zamkniętym – dane krążą w pętli, by zapewnić stabilność i wydajność systemu (stigmergia). Natomiast w certyfikacji – dane testowe są instrumentem kontroli (muszą być rzadkie i dobrze dobrane, inaczej test traci moc). **Utrzymanie wartości danych** wymaga więc różnych strategii: od ograniczania dostępu (by nie straciły unikatowości), przez ciągłe odświeżanie (benchmarki), po recykling i multiplikację (habitat regeneracyjny).
- **Ekonomia tych ekosystemów opiera się w dużej mierze na zarządzaniu ryzykiem.** W laboratorium społecznym zmniejsza się ryzyko społecznego odrzucenia technologii (co mogłoby pogrzebać najdroższy nawet projekt). W AI-ops fundamentalne jest zarządzanie ryzykiem finansowym (czy przychody pokryją olbrzymie koszty) i ryzykiem technicznym (awarie, ataki – stąd nacisk na nadmiarowość i bezpieczeństwo). Prototypownia dosłownie zajmuje się *wczesnym wykrywaniem ryzyk technologicznych*, działając niczym inwestor rozkładający ryzyko na portfolio małych zakładów – co zwiększa szanse trafienia. Habitat regeneracyjny mitiguje ryzyko braku dostaw z zewnątrz i awarii systemu podtrzymania życia przez autonomię i pętle sterowania (jednak tu pojawiają się inne ryzyka – np. nagromadzenie *długu technologicznego*, jeśli system jest zbyt skomplikowany do utrzymania bez resetu). Wreszcie, środowisko certyfikacji ma chronić przed ryzykiem katastrof z udziałem AI (np. certyfikowane AI w samochodzie ma zmniejszyć prawd. wypadku). **Rachunek opłacalności** w wielu z tych modeli nie jest trywialny, bo zyski mają często charakter pośredni lub długofalowy. Niemniej, metodologia np. Real Options czy ROI z programów innowacji ⁸ wskazuje, że inwestycja w kontrolowane środowiska testowe i adaptacyjne (jak te pięć ekosystemów) potrafi przynieść wielokrotny zwrot – choć wymaga to cierpliwości i odpowiedniej skali działania.
- **Wymiana wiedzy i synergiczne powiązania między modelami:** Warto zauważyć, że te ekosystemy nie muszą istnieć w izolacji – wręcz przeciwnie, mogą tworzyć **meta-ekosystem**. Np.

prototypownia może korzystać z danych z laboratorium społecznego, by tworzyć nowe usługi dla miasta; wyniki prototypów mogą zasilać AI-ops (gdy wejdą do produkcji jako nowe modele); habitat regeneracyjny może posłużyć jako poligon certyfikacyjny (np. testuje się tam AI w warunkach ekstremalnych przed certyfikacją do misji kosmicznych); laboratorium społeczne może adaptować modele wypracowane w AI-ops do lokalnych potrzeb itd. Takie powiązania zwiększają łączną wartość – np. **ISS National Lab współpracuje z prototypowniami firm poprzez programy SBIR**, testując ich technologie w mikrograwitacji ⁵¹; wyniki tamże mogą zostać skomercjalizowane (AI-ops) lub ustandaryzowane (certyfikacja astronautyczna). Innymi słowy, **ekonomia danych i kompetencji ma charakter sieciowy** – sukces rodzi się z koordynacji wielu ról i modeli, niczym dobrze funkcjonująca wioska, gdzie różne „gospodarstwa” (ekosystemy) wymieniają się plonami i usługami dla dobra całej społeczności.

- **Heurystyki sukcesu konvergują na kilka zasad wspólnych:** Po pierwsze, *adaptacyjność* – wszystkie modele muszą unikać stagnacji i uczyć się na doświadczeniach (czy to RLHF w AI-ops, czy retrospektywy w prototypach, czy ewolucja benchmarków). Po drugie, *utrzymanie równowagi pomiędzy wydajnością a bezpieczeństwem/uczciwością* – np. laboratorium społeczne dba by nie iść na skróty kosztem zaufania mieszkańców; AI-ops balansuje szybkość rozwoju a kontrolę (przykładowo, wprowadzanie mechanizmów alignementu choć to „podatek” na wydajność ⁵²); habitat regeneracyjny – równowaga między eksploatacją zasobów a regeneracją (by nie doprowadzić do załamania ekosystemu); certyfikacja – równowaga między innowacją a ostrożnością (zbyt rygorystyczne normy mogą hamować postęp, zbyt luźne – doprowadzić do wypadków). Po trzecie, *współpraca i otwartość na wiedzę zewnętrzną* – czy to włączanie mieszkańców, crowdsourcing, partnerstwa w przemyśle, standardy międzynarodowe – hermetyczność prowadzi do błędów strukturalnych (np. bias zamkniętej grupy), a otwartość – do *stigmergicznej optymalizacji w skali makro* (uczymy się od siebie nawzajem, jak mrówki, poprzez środowisko).

Na koniec warto podkreślić, że raport ten koncentrował się na **aspekcie modelowym i analitycznym** – każdy z opisanych ekosystemów istnieje (lub może istnieć) w różnych wariantach praktycznych. Ich studium ma znaczenie dla zrozumienia, jak budować **odpowiedzialne, efektywne i zrównoważone systemy human-AI**. W miarę postępu technologii i zmian społecznych, modele te będą się zapewne ewoluować. Jednak fundamenty – czyli spojrzenie na ekosystem AI nie tylko technicznie, ale holistycznie: jako na *habitat* z ekonomią, społecznymi interakcjami, cyklem życia i ryzykiem – pozostaną istotne dla projektowania przyszłości, w której ludzie i sztuczna inteligencja tworzą wspólną, korzystną symbiozę.

¹ ⁴³ ⁴⁵ Wakacje w kosmosie? - coraz bardziej realne turystyczne loty kosmiczne - Dziennik Warto Wiedzieć

<https://wartowiedziec.pl/serwis-glowny/styl-zycia/53900-wakacje-w-kosmosie-coraz-bardziej-realne-turystyczne-loty-kosmiczne>

² ³ ⁴ ⁵ ⁶ ⁷ ¹³ ⁴⁸ Rotterdam launches Social AI Lab for inclusive AI - RotterdamStyle.com

<https://rotterdamstyle.com/government/rotterdam-launches-social-ai-lab-for-inclusive-ai>

⁸ ³⁴ ³⁵ ³⁶ Microsoft Word - Impacts of the SBIR-STTR programs 05_29_2017

<https://sbtc.org/wp-content/uploads/2018/02/Impacts-of-the-SBIR-program.pdf>

⁹ FAIR data principles - National Library of Medicine

<https://www.nlm.nih.gov/oet/ed/cde/tutorial/02-200.html>

¹⁰ Inteligencja roju: Jak kolektywna mądrość zmienia świat technologii – tematydla wszystkich.pl

<https://tematydla wszystkich.pl/2025/09/07/inteligencja-roju-jak-kolektywna-madrosz-zmienia-swiat-technologii/>

- 11 12 28 52 **Aligning language models to follow instructions | OpenAI**
<https://openai.com/index/instruction-following/>
- 14 18 24 31 47 **OpenAI CFO Sarah Friar Envisions New Licensing Models for AI Revenue - Business Insider**
<https://www.businessinsider.com/openai-cfo-sarah-friar-future-revenue-sources-2026-1>
- 15 **How Data Monetization & Data Modernization Can Improve Business ROI**
<https://www.materialplus.io/perspectives/from-data-modernization-to-data-monetization-make-the-play-for-better-business-roi>
- 16 40 41 **Winning SBIR grants for your climate business**
<https://enduringplanet.com/insights/winning-sbir-grants>
- 17 **Chiński model AI trafił na orbitę Ziemi. Czy polskie firmy są gotowe na nowy wyścig kosmiczny? (WYWIAD) - XYZ**
<https://xyz.pl/chinski-model-ai-trafil-na-orbite-ziemi-czy-polskie-firmy-sa-gotowe-na-nowy-wyscig-kosmiczny-wywiad/>
- 19 **OpenAI Revenue 2024: Complete ARR, API vs ChatGPT Breakdown**
<https://futuresearch.ai/openai-revenue-report/>
- 20 **Maxar's Sentry Is the Satellite System Governments Have Been ...**
<https://orbitaltoday.com/2025/06/27/maxars-sentry-is-the-satellite-system-governments-have-been-waiting-for/>
- 21 **Satya Nadella Said Bill Gates Warned Against Investing in OpenAI**
<https://www.businessinsider.com/microsoft-satya-nadella-bill-gates-warn-invest-openai-billion-burn-2025-10>
- 22 25 46 49 **ChatGPT's hidden toll: AI courtesy fuels \$700K daily costs**
<https://www.ynetnews.com/business/article/bkiagmekxx>
- 23 **If chatgpt costs \$700000 to run per DAY, how do they not ... - Reddit**
https://www.reddit.com/r/ChatGPT/comments/1c5ri5o/if_chatgpt_costs_700000_to_run_per_day_how_do/
- 26 27 44 **Investment Perspectives: Opportunity and Economics of ISS National Lab Research**
<https://issnationallab.org/iss360/investment-perspectives-opportunity-and-economics-of-iss-national-lab-research/>
- 29 **SecureWatch by Maxar | Esri Partner Solution**
<https://www.esri.com/partners/maxar-a2T70000000TNOvEAO/securewatch-a2d5x000005kI0KAAU>
- 30 **A Complete Guide to LLM Benchmark Categories - Galileo AI**
<https://galileo.ai/blog/llm-benchmarks-categories>
- 32 51 **About Flight Opportunities - NASA**
<https://www.nasa.gov/stmd-flight-opportunities/about-flight-opportunities/>
- 33 **14 Readiness Level Frameworks: The Guide to TRL, MRL, SRL, and ...**
<https://www.itonics-innovation.com/blog/14-readiness-level-frameworks>
- 37 42 **ESA - MELiSSA life support project, an innovation network in support to space exploration**
https://www.esa.int/Enabling_Support/Space_Engineering_Technology/MELiSSA_life_support_project_an_innovation_network_in_support_to_space_exploration
- 38 **Technology readiness levels - Canada.ca**
<https://ised-isde.canada.ca/site/innovation-canada/en/technology-readiness-levels>
- 39 **Technology Readiness Level (TRL): Guide 1-9 - BionIT Labs**
<https://bionitlabs.com/tech-en/trl-technological-development-adams-hand/>
- 50 **International Space Station Open-Source Data - PMC**
<https://pmc.ncbi.nlm.nih.gov/articles/PMC7733869/>