

● APPLICATION FIRST

Manual strategii sprzedaży doradczej

Od produktu do rozwiązania problemów klienta

1. CEL I ZAKRES MANUALA

Celem niniejszego manuala jest **operacjonalizacja strategii „Application First”** w codziennej pracy zespołów sprzedaży i wsparcia technicznego **Milwaukee**.

Dokument:

- definiuje **czym Application First jest, a czym nie jest**,
 - opisuje **rolę JSS jako kluczowego nośnika modelu**,
 - wprowadza **Application First Canvas** jako **standard wizyty terenowej**,
 - zamienia strategię z hasła w **powtarzalny proces operacyjny**.
-

2. ISTOTA STRATEGII APPLICATION FIRST

Application First to podejście sprzedażowe skoncentrowane na **rozwiązaniu realnego problemu biznesowego lub technologicznego klienta**, poprzez zrozumienie **konkretnej aplikacji (zastosowania)** narzędzia w jego codziennej pracy.

W tym modelu:

- **produkty są środkiem**, nie celem,
- **aplikacja i proces pracy klienta** są punktem wyjścia,
- wartość definiowana jest **językiem klienta**, a nie parametrami technicznymi.

Application First odpowiada na zmianę rynku:

klienci nie szukają już „lepszych narzędzi”, lecz **lepszych sposobów wykonywania pracy**.

3. FUNDAMENT MODELU: DIAGNOZA I WARTOŚĆ

Podstawą Application First jest **diagnoza pracy klienta**, a nie prezentacja produktu.

3.1. Zmiana punktu startu rozmowy

Podejście produktowe Podejście Application First

„Ile ma obrotów?”

„Gdzie i jak pracujesz?”

„Nowość w ofercie”

„Co dziś Cię spowalnia?”

„Najmocniejszy silnik”

„Gdzie tracisz czas / energię?”

3.2. Cel diagnozy

Celem nie jest opis produktu, lecz:

- zrozumienie **warunków pracy**,
 - identyfikacja **ograniczeń procesu**,
 - przełożenie cech rozwiązania na **mierzalną wartość**:
 - oszczędność czasu,
 - wzrost bezpieczeństwa,
 - poprawę wydajności,
 - redukcję kosztów.
-

4. ROLA SPECJALISTÓW JSS (JOB SITE SOLUTIONS)

Specjaliści JSS są **kluczowym ogniwem strategii Application First**.

4.1. Rola JSS w modelu

JSS:

- pracuje **bezpośrednio w miejscu wykonywania pracy** (jobsite, warsztat),
- obserwuje realne zastosowanie narzędzi,
- łączy **produkt, aplikację i użytkownika końcowego**,
- pełni rolę **doradcy technicznego**, nie sprzedawcy katalogowego.

4.2. JSS jako „tłumacz wartości”

JSS:

- tłumaczy język technologii na język pracy,
- przekłada cechy systemu na **konkretne efekty operacyjne**,
- prowadzi klienta przez proces decyzyjny oparty na faktach z aplikacji.

5. FUNKCJA INTEGRACYJNA APPLICATION FIRST

Application First jest **wspólnym językiem organizacyjnym**, który:

- łączy **Sell-out (użytkownik)** i **Sell-in (dystrybutor)**,
- spina działania **sprzedaży, marketingu i trade marketingu**,
- zapobiega silosowości i sprzecznym komunikatom.

Model ten przesuwaa organizację:

- z koncentracji na **wewnętrznych celach produktowych**,
- na koncentrację na **realnych problemach rynku**.

6. ROZDZIEWIĘK MIĘDZY STRATEGIĄ A PRAKTYKĄ

(wnioski audytowe)

Audyt wykazał, że mimo strategicznego znaczenia Application First:

6.1. Główne luki wdrożeniowe

- wizyty często zaczynają się od **prezentacji nowości**, a nie diagnozy,
- demo bywa **technologiczne**, a nie aplikacyjne,
- brak jest **wspólnego rozumienia**, czym Application First jest w praktyce,
- w części zespołów presja rotacji i złożoność oferty ograniczają jakość rozmowy.

6.2. Skutek

Application First funkcjonuje często jako **hasło**, a nie jako:

- standard zachowań,
- powtarzalny proces,
- mierzalny model pracy w terenie.

Odpowiedzią na te luki jest **Application First Canvas**.

7. METAFORA OPERACYJNA

Application First można porównać do pracy **dobrego lekarza**:

- podejście produktowe → aptekarz („mamy nową tabletkę”),

- Application First → lekarz:
 1. bada pacjenta (analiza aplikacji),
 2. diagnozuje źródło bólu (problem),
 3. tłumaczy skutki braku leczenia (konsekwencje),
 4. dobiera leczenie (system),
 5. sprawdza efekt (demo),
 6. omawia poprawę stanu (wartość),
 7. ustala dalsze kroki (follow-up).
-

8. APPLICATION FIRST CANVAS

Standard wizyty terenowej (7 kroków)

Application First Canvas to **obowiązująca mapa rozmowy JSS**.

Każdy etap ma:

- jasno określony **cel**,
 - oczekiwane **zachowania**,
 - kryterium **sukcesu**.
-

KROK 1: APLIKACJA (Job to be Done)

Cel: zrozumienie realnej pracy klienta

Zasada: bez rozmów o produkcie

Sukces: JSS rozumie proces pracy klienta

KROK 2: PROBLEM (Pain Points)

Cel: wydobyć realnych problemów

Zasada: klient sam nazywa ból

Sukces: klient werbalizuje problem

KROK 3: KONSEKWENCJE (Impact)

Cel: przeliczenie problemu na straty
Zasada: klient sam kwantyfikuje
Sukces: klient widzi koszt braku zmiany

KROK 4: ROZWIĄZANIE MILWAUKEE (System)

Cel: dopasowany system, nie katalog
Zasada: narzędzie + bateria + akcesoria
Sukces: rozwiązanie odpowiada na zdiagnozowany ból

KROK 5: DEMO APLIKACYJNE (Proof)

Cel: udowodnienie rozwiązania w pracy
Zasada: użytkownik pracuje narzędziem
Sukces: klient czuje różnicę w aplikacji

KROK 6: WARTOŚĆ (Value)

Cel: klient sam wylicza wartość
Zasada: brak narzucania ROI
Sukces: wartość opisana językiem klienta

KROK 7: NEXT STEPS (Domknięcie)

Cel: jasne zobowiązania
Standard: Kto – Co – Kiedy + follow-up ≤24h
Sukces: ustalone dalsze działania

9. ROLA CANVASU W ORGANIZACJI

Application First Canvas:

- jest **wspólnym standardem** dla JSS, Sell-out i Sell-in,
 - stanowi punkt odniesienia dla audytu i coachingu,
 - zamienia improwizację w **przewidywalny mechanizm operacyjny**.
-

10. PODSUMOWANIE

Application First to nie technika sprzedaży.

To **zmiana sposobu myślenia organizacji**:

z „wciskania pudełek”

na **doradztwo oparte na zrozumieniu pracy klienta**.

7 kroków Application First Canvas wplecione w logikę „Systemu Milwaukee”

rozumianego jako 4 elementy:

SYSTEM = (1) NARZĘDZIE + (2) PLATFORMA (bateria/ekosystem) + (3) OSPRZĘT + (4) OCHRONA (PPE/ergonomia/bezpieczeństwo)

Każdy krok ma: **CEL → DZIAŁANIA → JAK WPLEŚĆ SYSTEM (4 elementy) → PRZYKŁAD**.

● APPLICATION FIRST CANVAS (SYSTEM-DRIVEN)

7 kroków z prezentacją systemu Milwaukee

1) APLIKACJA (Job to be Done)

Cel

Zrozumieć **pracę klienta w realnym środowisku**: co robi, jak robi, w jakich warunkach.

Działania

- obserwacja na miejscu (jobsite / warsztat / hala)
- mapowanie czynności: *krok po kroku* (co, gdzie, jak, jak często)
- zbieranie danych o warunkach: materiał, dostęp, pył, wysokość, wilgoć, hałas, czas

Jak wpleść SYSTEM (bez mówienia o produktach)

Tu **nie sprzedajesz systemu** — ale zbierasz „parametry aplikacji”, które potem pozwolą zbudować 4 elementy:

- **narzędzie**: typ pracy (wiercenie/cięcie/mocowanie)
- **platforma**: czas pracy, mobilność, przerwy, dostęp do zasilania
- **osprzęt**: materiał, jakość cięcia/otworu, powtarzalność
- **ochrona**: ryzyka (pył, hałas, cięcie, wysokość), ergonomia

Przykład

„Robimy kotwienie pod instalacje pod sufitem, beton zbrojony, 80–120 otworów dziennie, nad głową, często na drabinie.”

2) PROBLEM (Pain Points)

Cel

Doprowadzić do tego, by klient **sam nazwał ból** (czas / ergonomia / bezpieczeństwo / koszty).

Działania

- pytania pogłębiające + parafraza
- wydobywanie „momentów tarcia” w aplikacji
- zakaz sugerowania problemu (klient musi go wypowiedzieć)

Jak wpleść SYSTEM

Na tym etapie **problem przypinasz do elementu systemu**, ale nadal nie proponujesz rozwiązań:

- ból = *za wolno / przegrzewanie / brak mocy* → potencjalnie **narzędzie**
- ból = *przerwy na ładowanie / brak baterii / ciężar* → **platforma**
- ból = *wiertła się palą / tarcze słabe / dużo poprawek* → **osprzęt**
- ból = *zmęczenie / ryzyko / pył / hałas / urazy* → **ochrona**

Przykład

Klient: „Nad głową po 2 godzinach ręce siadają. A jak wiertło złapie zbrojenie, to wszystko staje.”

3) KONSEKWENCJE (Impact)

Cel

Przeliczyć problemy na **straty operacyjne**: czas, koszty, ryzyko, jakość.

Działania

- klient sam kwantyfikuje („ile razy”, „ile minut”, „ile poprawek”)
- przeliczenie strat na tydzień / miesiąc / projekt
- pokazanie kosztu „pozostania przy starym sposobie pracy”

Jak wpleść SYSTEM

Konsekwencje mapujesz na system, żeby w kroku 4 zbudować celowane rozwiązanie:

- **narzędzie** → straty czasu przez brak mocy/tempo
- **platforma** → przerwy, logistyka baterii, downtime
- **osprzęt** → poprawki, zużycie, zniszczenia materiału
- **ochrona** → ryzyko wypadku, absencja, przestoje, compliance

Przykład

Klient: „Jak trafiamy na zbrojenie, tracę średnio 20 minut na serię. Do tego 2–3 wiertła tygodniowo do kosza.”

4) ROZWIĄZANIE MILWAUKEE (System)

Cel

Zaproponować **system rozwiązania**, nie pojedynczy produkt i nie „katalog”.

Działania

- budujesz rozwiązanie jako „receptę” na 4 bóle (jeśli występują)
- pokazujesz logikę: „*to odpowiada na to*”
- ograniczasz wybór do minimum sensownego (2 opcje max)

SYSTEM (4 elementy) — standard prezentacji

Tu wchodzisz w jasny schemat:

1. **NARZĘDZIE** – „co robi robotę”
2. **PLATFORMA** – „co zapewnia ciągłość” (bateria/ekosystem)
3. **OSPRZĘT** – „co dotyka materiału i robi różnicę”
4. **OCHRONA** – „co zabezpiecza człowieka i jakość pracy”

Przykład

„Jeśli kluczowy jest tempo i praca nad głową:

- narzędzie: lżejsze / o wyższej wydajności w betonie
- platforma: bateria dająca ciągłość bez spadku mocy
- osprzęt: wiertło do zbrojonego, które nie pali się na pręcie

- ochrona: okulary + rękawice + element ergonomii chwytu”
-

5) DEMO APLIKACYJNE (Proof)

Cel

Udowodnić w realnej aplikacji, że **system** rozwiązuje problem.

Działania

- ustalasz kryterium sukcesu demo (konkret!)
- klient pracuje narzędziem — JSS prowadzi
- w demo testujesz **system**, nie samą maszynę

Jak prowadzić demo systemowe (4 elementy)

W demo **pokazujesz zależność**:

- **narzędzie**: prędkość / brak przegrzania / kontrola
- **platforma**: stała moc, brak przestojów, balans
- **osprzęt**: jakość otworu/cięcia, brak przypaleń, mniej docisku
- **ochrona**: mniej pyłu w twarz, stabilniejszy chwyt, mniejsze ryzyko

Klucz: nie mówisz „to jest lepsze”. Pytasz: „co widzisz i czujesz?”

Przykład

Kryterium: „Zrób 20 otworów nad głową bez przerw i bez spadku tempa. Sprawdź, czy musisz dociskać.”

Klient: „Idzie szybciej i mniej mnie wrywa, wiertło robi robotę.”

6) WARTOŚĆ (Value)

Cel

Klient sam opisuje wartość — w swoim języku.

Działania

- pytania otwarte o zmianę
- klient liczy (czas, koszt, komfort, bezpieczeństwo)
- zakaz wciskania ROI (to klient ma „wziąć własność”)

✖ Jak przypisać wartość do 4 elementów systemu

Pomagasz klientowi „nazwać wartość” w czterech wymiarach:

- **narzędzie:** „szybciej”, „pewniej”, „mniej poprawek”
- **platforma:** „mniej przerw”, „jedna bateria mniej”, „lżejszy set”
- **osprzęt:** „mniej zużycia”, „mniej spalonych wiertel”, „czystsze cięcie”
- **ochrona:** „mniej ryzyka”, „mniej zmęczenia”, „bezpieczniej na drabinie”

Przykład

Klient: „Jak odzyskam 30 minut dziennie, to w miesiącu mam prawie dzień pracy. I mniej wiertel spalam — realnie oszczędzam.”

7) NEXT STEPS (Domknięcie)

Cel

Przekuć wartość i decyzję techniczną w **konkretny plan działania**.

Działania

- ustalasz KTO–CO–KIEDY
- potwierdzasz proces zakupu (sell-in / dystrybutor / KAM)
- follow-up ≤ 24h (podsumowanie + system + ustalenia)

✖ Jak domykać „system”, a nie „produkt”

W next steps nie zamykasz „wiertarki”. Zamykasz:

- **system** (narzędzie + platforma + osprzęt + ochrona)
- plus warunki testu / dostępność / oferta / zamówienie

Przykład

„Kto: ja + Wasz brygadzysta

Co: test zestawu systemowego 1 tydzień + oferta na komplet

Kiedy: wracam w piątek 10:00, a follow-up dostajesz dziś do 16:00.”

Mini-podsumowanie (zdanie do manuala)

Application First = diagnozujesz pracę → klient nazywa ból → liczysz koszt → proponujesz system → udowadniasz w demo → klient nazywa wartość → ustalacie next steps.

Zestaw DO / DON'T dla 7 kroków Application First Canvas, zintegrowany z logiką SYSTEMU
(narzędzie / platforma / osprzęt / ochrona).

APPLICATION FIRST — DO / DON'T





Standard zachowań w terenie – wersja bez kompromisów
Milwaukee


APLIKACJA (Job to be Done)

DO

- obserwuj **realną pracę** klienta (patrz, nie zgaduj)
- pytaj o **co, jak, jak często, w jakich warunkach**
- notuj elementy pod przyszły **system** (warunki → narzędzie / platforma / osprzęt / ochrona)
- słuchaj **języka klienta**, nie poprawiaj go technicznie

DON'T

-  nie mów o produktach, nowościach, promocjach
-  nie zakładaj aplikacji na podstawie branży („elektryk = to”)
-  nie pokazuj katalogu „żeby zobaczyć”
-  nie skracaj etapu „bo klient się spieszy”

 **Błąd krytyczny:** przechodzenie do produktu bez zrozumienia pracy

2 PROBLEM (Pain Points)

DO

- doprowadź do tego, aby **klent sam nazwał problem**
- eksploruj 4 obszary:
 - ergonomia
 - czas
 - bezpieczeństwo
 - koszty / poprawki
- parafrazuj („czyli największym problemem jest...”)

❌ DON'T

- ❌ nie sugeruj problemu („często klienci mówią, że...”)
- ❌ nie diagnozuj za klienta
- ❌ nie minimalizuj bólu („to normalne”)
- ❌ nie przechodź dalej bez werbalizacji problemu

👉 **Zasada żelazna:** *jeśli klient nie nazwie problemu – nie ma decyzji*

3 KONSEKWENCJE (Impact)

✅ DO

- każ klientowi **samemu liczyć straty**
- przelicz:
 - czas → pieniądz
 - zmęczenie → tempo
 - ryzyko → przestoje
- dopytuj: „ile razy?”, „ile minut?”, „ile w tygodniu?”

❌ DON'T

- ❌ nie licz za klienta
- ❌ nie używaj „średnich rynkowych”
- ❌ nie strasz wypadkami
- ❌ nie pomijaj etapu „bo wszyscy wiedzą, że to kosztuje”

👉 **Błąd typowy:** przejście do rozwiązania bez policzonych konsekwencji

4 ROZWIĄZANIE MILWAUKEE (SYSTEM)

✅ DO

- prezentuj **SYSTEM**, zawsze w tej kolejności:
 1. narzędzie
 2. platforma (bateria / ekosystem)
 3. osprzęt
 4. ochrona (PPE / ergonomia)
- pokaż **logikę dopasowania**: „to odpowiada na ten problem”
- ogranicz wybór do **maks. 1–2 sensownych opcji**

❌ DON'T

- ❌ nie pokazuj całej oferty
- ❌ nie zaczynaj od ceny
- ❌ nie sprzedawaj „samej maszyny”
- ❌ nie rób cross-sellu bez powiązania z problemem

👉 **Błąd systemowy:** produkt zamiast rozwiązania

5 DEMO APLIKACYJNE (Proof)

✅ DO

- ustal **konkretne kryterium sukcesu demo**
- oddaj narzędzie **w ręce klienta**
- testuj **cały system**, nie tylko maszynę
- pytaj: „co czujesz?”, „co widzisz?”

❌ DON'T

- ❌ nie rób „pokazu z ręki JSS”
- ❌ nie gadaj w trakcie pracy klienta

- ❌ nie demonstruj funkcji bez kontekstu
- ❌ nie rób demo bez celu („zobaczmy jak pójdzie”)

👉 **Błąd klasyczny:** demo technologiczne zamiast aplikacyjnego

6 WARTOŚĆ (Value)

✅ DO

- pozwól klientowi **samemu nazwać wartość**
- zadawaj pytania otwarte:
 - „co się zmieniło?”
 - „co to daje Tobie?”
- zbieraj wartość w 4 wymiarach systemu:
 - tempo / ciągłość / zużycie / bezpieczeństwo

❌ DON'T

- ❌ nie wciskaj ROI
- ❌ nie pokazuj excela „na siłę”
- ❌ nie porównuj do konkurencji ceną
- ❌ nie kończ na „fajnie się pracuje”

👉 **Zasada:** wartość klienta > kalkulacja handlowca

7 NEXT STEPS (Domknięcie)

✅ DO

- zawsze kończ wizytę formułą **KTO – CO – KIEDY**
- domykaj **SYSTEM**, nie pojedynczy produkt
- wysyłaj follow-up ≤ **24h**
- jasno wskaż kolejny krok (test / oferta / zamówienie)

❌ DON'T

- ❌ nie kończ wizyty bez ustaleń

- ❌ nie mów „odezwę się”
- ❌ nie zostawiaj klienta bez podsumowania
- ❌ nie odkładaj follow-upu „na jutro”

👉 **Błąd końcowy:** dobra rozmowa bez dalszego procesu

● PODSUMOWANIE ZACHOWAŃ KRYTYCZNYCH

Application First upada, gdy:

- zaczynasz od produktu
- robisz demo bez celu
- liczysz wartość za klienta
- nie domykasz next steps

Application First działa, gdy:

- diagnozujesz pracę
 - klient sam nazywa ból i koszt
 - prezentujesz system
 - udowadniasz w aplikacji
 - klient sam mówi „to ma sens”
-