

# RAPORT-PRÓBKA - przygotowany na podstawie syntetycznego workflow, zanonimizowana struktura, rzeczywista metodologia

## Loop Autopsy: przyjęcie zamówienia, synchronizacja realizacji, odświeżanie KPI i dzienny raport

|                     |                                                                                                                                                                                                                                              |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Klient              | Syntetyczny sprzedawca internetowy                                                                                                                                                                                                           |
| Audytowy proces     | Od webhooka zamówienia do realizacji oraz odświeżanie marketingowych KPI i dzienny raport                                                                                                                                                    |
| Platforma           | n8n, eksport syntetyczny, tryb hostingu nie był oceniany                                                                                                                                                                                     |
| Wersja workflow     | Storefront Order Sync v7 i Marketing KPI Refresh v4, eksport 2026-08-01                                                                                                                                                                      |
| Cel biznesowy       | Gdy przychodzi prawidłowe zamówienie, system ma utworzyć dokładnie jeden rekord realizacji i zachować spójny stan klienta oraz sklepu. Co 15 minut ma odświeżyć KPI, a o 08:00 Europe/Warsaw wysłać raport zawierający oba źródła reklamowe. |
| W zakresie          | 2 workflow, 30 nodów, 700 wykonania, jedna główna ścieżka każdego workflow, okres 2026-07-25 do 2026-07-31                                                                                                                                   |
| Poza zakresem       | Dane uwierzytelniające, administracja platformą, reguły bramy przed webhookiem, konfiguracja dostawców, infrastruktura, billing i jakiegokolwiek żywe środowisko                                                                             |
| Otrzymane materiały | 2 syntetyczne eksporty n8n JSON, 28 wykonania synchronizacji zamówień, 672 wykonania marketingowe, deterministyczny generator historii                                                                                                       |
| Data raportu        | 2026-08-01                                                                                                                                                                                                                                   |
| Metoda              | Stała lista 25 kontroli w pięciu obszarach, każde finalne ustalenie przypięte do linii źródła i konkretnego runu                                                                                                                             |

### 1. Podsumowanie zarządcze

Oba workflow kończą większość rutynowych wykonania, ale zielony status nie jest wiarygodnym dowodem poprawnego wyniku biznesowego. Jeden nieprawidłowy webhook dotarł do wysyłki potwierdzenia, zanim jego podpis został odrzucony. Jeden łańcuch timeoutu i retry utworzył dwa rekordy realizacji dla tego samego syntetycznego zamówienia, ponieważ POST nie miał ani klucza idempotencji, ani przerwy pomiędzy próbami. Workflow marketingowy miał 5 nakładających się startów, osiągnął współbieżność 3 i 30 lipca zapisał starsze okno KPI po nowszym. Dwa z 7 dziennych raportów wysłano z tylko jednym z dwóch oczekiwanych źródeł reklamowych, ponieważ po błędzie HTTP proces kontynuował pracę. Konfiguracja i wyniki nazwanych runów są zgodne, dlatego pewność pięciu ustaleń jest wysoka, natomiast kontrole na poziomie platformy pozostają nieznane. Natychmiastowa decyzja to umieszczenie walidacji i ochrony przed duplikatem przed każdą nieodwracalną akcją, zanim wystartuje kolejny ryzykowny run.

**Ogólne ryzyko:** CRITICAL

**Pewność diagnozy:** wysoka dla pięciu ustaleń, ograniczona dla kontroli niewidocznych w eksportach.

**Jedna najważniejsza decyzja:** Przenieść kontrolę podpisu przed pierwszy email i wymagać stabilnego klucza idempotencji przed utworzeniem realizacji.

### 2. Scorecard

| Obszar                                  | Ocena        | Ustalenia według severity | Jedno zdanie                                                                                  |
|-----------------------------------------|--------------|---------------------------|-----------------------------------------------------------------------------------------------|
| <b>A. Triggery i harmonogram</b>        | FAIL         | C:1 H:1 M:0 L:0           | Nieprawidłowe żądanie wywołało email, a 5 startów harmonogramu nałożyło się na siebie.        |
| <b>B. Obsługa błędów i odzyskiwanie</b> | FAIL         | C:1 H:1 M:0 L:0           | Retry zduplikował rekord u dostawcy, a jedno wywołanie HTTP może czekać bez timeoutu noda.    |
| <b>C. Cichy drift i jakość wyniku</b>   | FAIL         | C:0 H:1 M:0 L:0           | Dwa dzienne raporty opublikowano bez jednego oczekiwanego źródła.                             |
| <b>D. Koszt i pętle</b>                 | NOT ASSESSED | C:0 H:0 M:0 L:0           | Nie dostarczono billingu platformy ani dostawców.                                             |
| <b>E. Obserwowalność i backup</b>       | NOT ASSESSED | C:0 H:0 M:0 L:0           | Eksporty nie dowodzą istnienia zewnętrznego canary, właściciela alertu ani próby odtworzenia. |

### 3. Ustalenia

#### 3.1 Indeks ustaleń

| ID   | Obszar | Ustalenie                                                              | Severity | Status dowodu | Nakład |
|------|--------|------------------------------------------------------------------------|----------|---------------|--------|
| F-01 | A      | Potwierdzenie jest wysyłane przed walidacją webhooka                   | CRITICAL | CONFIRMED     | S      |
| F-02 | B      | Retry realizacji nie ma klucza idempotencji                            | CRITICAL | CONFIRMED     | M      |
| F-03 | A      | Zaplanowane odświeżenia KPI nakładają się i zapisują w złej kolejności | HIGH     | CONFIRMED     | M      |
| F-04 | C      | Obsłużony błąd źródła reklamowego publikuje niepełny raport            | HIGH     | CONFIRMED     | M      |
| F-05 | B      | Wywołanie Meta Ads nie ma timeoutu na poziomie noda                    | HIGH     | CONFIRMED     | S      |

#### F-01. Potwierdzenie przed walidacją webhooka

**Obszar:** A. Triggery i harmonogram

**Punkt checklisty:** T1, T5

**Severity:** CRITICAL

**Status dowodu:** CONFIRMED

**Co zaobserwowałem**

Webhook jawnie używa `authentication: none` w `order-sync-v7-synthetic.json:23`. Graf prowadzi następnie

przez nod `Send Order Acknowledgement` przed nodem `Validate Webhook Signature`, co widać w liniach 44, 59 i 247. W runie `0S-20260729-0142`, zaczynającym się w `order-sync-runs-7d.json:1962`, wynik podpisu to `invalid` w linii 2020, ale przed ścieżką odrzucenia zapisano jeden wysłany email potwierdzający.

#### **Dlaczego to ma znaczenie**

Odrzucenie następuje za późno. Nieprawidłowe żądanie może wywołać nieodwracalną wiadomość zewnętrzną, mimo że zamówienie nie zostanie przyjęte.

#### **Naprawa**

1. Przenieść `Validate Webhook Signature` bezpośrednio za webhook.
2. Odrzucać błędny podpis przed normalizacją i side effectami.
3. Pozostawić potwierdzenie na prawidłowej gałęzi i liczyć odrzucenia bez payloadów.

#### **Jak zweryfikować naprawę**

W kopii sandboxowej wysłać jeden prawidłowy fixture i jeden fixture z błędnym podpisem. Nieprawidłowy przypadek musi zwrócić odrzucenie, wysłać zero emaili, utworzyć zero realizacji i zero rezerwacji zapasu. Prawidłowy przypadek nadal ma wysłać jedno potwierdzenie.

**Owner:** opiekun workflow | **Nakład:** S, poniżej 1 godziny

### **F-02. Retry realizacji bez idempotencji**

**Obszar:** B. Obsługa błędów i odzyskiwanie

**Punkt checklisty:** E2, E3

**Severity:** CRITICAL

**Status dowodu:** CONFIRMED

#### **Co zaobserwowałem**

Nod `Create Fulfilment Order` włącza 3 próby bez przerwy w `order-sync-v7-synthetic.json:103` oraz liniach 107 do 109. Nagłówki nie zawierają klucza idempotencji ani deduplikacji, a obiekt `options` jest pusty w linii 118. W runie `0S-20260728-0807`, zaczynającym się w `order-sync-runs-7d.json:1578`, pierwsza próba zapisuje timeout klienta po przyjęciu żądania przez dostawcę w linii 1628. Retry kończy się sukcesem z drugim identyfikatorem dostawcy, a uzgodnienie stanu docelowego pokazuje 2 rekordy realizacji w linii 1704.

#### **Dlaczego to ma znaczenie**

Workflow nie odróżnia błędu przed przyjęciem od przyjęcia z utraconą odpowiedzią. Retry powtarza POST i może zmienić dwa rekordy dostawcy w dwie wysyłki albo rezerwacje.

#### **Naprawa**

1. Przed POST obliczyć i zapisać stabilny klucz, na przykład `order_id + fulfilment_v1`.
2. Użyć mechanizmu idempotencji dostawcy i wyszukać ten klucz przed ponownym utworzeniem.
3. Powtarzać tylko timeouty, 429 i wybrane 5xx, z przerwami 10, 60 i 180 sekund.
4. Po ostatniej próbie zatrzymać zamówienie zamiast kontynuować.

#### **Jak zweryfikować naprawę**

Wymusić timeout odpowiedzi po przyjęciu zamówienia sandboxowego przez dostawcę. W całym ciągu pierwszego wywołania i retry musi istnieć dokładnie jeden rekord dostawcy dla stabilnego klucza. Drugie wykonanie z tym samym order ID ma zwrócić lub wykorzystać ten sam rekord.

**Owner:** opiekun workflow i właściciel API realizacji | **Nakład:** M, od 1 do 4 godzin

### **F-03. Nakładające się odświeżenia KPI zapisują w złej kolejności**

**Obszar:** A. Triggery i harmonogram

**Punkt checklisty:** T3, T4

**Severity:** HIGH

**Status dowodu:** CONFIRMED

#### **Co zaobserwowałem**

Nod `Every 15 Minutes` jest triggerem harmonogramu w `marketing-kpi-refresh-synthetic.json:16`. Historia

672 runów zawiera 5 startów w chwili, gdy poprzedni run nadal trwał, a maksymalna współbieżność wyniosła 3. Dnia 30 lipca run MK-20260730-0815 zaczyna się w `marketing-report-runs-7d.json:49861` i zapisuje snapshot o 08:18:34Z, linia 49953. Run starszego okna MK-20260730-0800, zaczynający się w linii 49764, zapisuje o 08:19:40Z, linia 49856. Starsze okno trafia więc do celu 66 sekund po nowszym.

#### **Dlaczego to ma znaczenie**

Jeżeli dashboard uznaje ostatni zapis za aktualny, może cofnąć się w czasie przy zielonych wykonaniach. To zdarzyło się w dostarczonej historii.

#### **Naprawa**

1. Dodać blokadę jednego aktywnego runu albo ustawić concurrency na 1.
2. Wykonywać upsert każdego okna po unikalnym kluczu.
3. Odrzucać stare zapisy i kierować opóźnione okna do replay.

#### **Jak zweryfikować naprawę**

Uruchomić wolny fixture 08:00 i rozpocząć fixture 08:15 zgodnie z harmonogramem. Oba okna mogą się zakończyć, ale bieżący wiersz dashboardu musi pozostać 08:15, dla każdego okna ma istnieć jeden rekord, a decyzja lock albo replay ma być widoczna.

**Owner:** opiekun workflow | **Nakład:** M, od 1 do 4 godzin

### **F-04. Obsłużony błąd źródła publikuje niepełny raport**

**Obszar:** C. Cichy drift i jakość wyniku

**Punkt checklisty:** D1, O2

**Severity:** HIGH

**Status dowodu:** CONFIRMED

#### **Co zaobserwowałem**

Nod Fetch Google Ads ma `continueOnFail: true` w `marketing-kpi-refresh-synthetic.json:52` i linii 56. Jego ścieżka nadal prowadzi do `Append KPI Snapshot` oraz gałęzi dziennego raportu w liniach 175, 179 i 180. Historia zawiera 3 niepełne runy KPI. Dwa z nich były raportami o 08:00 Europe/Warsaw: MK-20260727-0600 w `marketing-report-runs-7d.json:20990` i MK-20260731-0600 w linii 58300. Oba zapisują 1 uwzględnione źródło przy 2 oczekiwanych, na przykład w liniach 21095 i 21098, a status wykonania pozostaje `success`. To 2 niepełne raporty z 7 dziennych raportów.

#### **Dlaczego to ma znaczenie**

Raport wygląda na kompletny, ale pomija jedną platformę reklamową. Decyzja o wydatkach, przychodzie albo ROAS może być błędna bez czerwonego runu.

#### **Naprawa**

1. Skierować błąd do gałęzi held, która nazywa niedostępne źródło.
2. Wymagać oczekiwanego zestawu źródeł w `Validate KPI Completeness`.
3. Blokować normalny raport przy braku źródła, wysłać czytelny alert i zatrzymać do replay.

#### **Jak zweryfikować naprawę**

Podać zapis odpowiedzi Google Ads 503 do runu sandboxowego. Normalny raport nie może zostać wysłany, kompletny wiersz KPI nie może zostać zapisany, a zatrzymany element ma nazywać brakujące źródło. Gdy oba fixture są dostępne, ma zostać wysłany jeden kompletny raport.

**Owner:** właściciel marketing operations i opiekun workflow | **Nakład:** M, od 1 do 4 godzin

### **F-05. Wywołanie Meta Ads bez timeoutu noda**

**Obszar:** B. Obsługa błędów i odzyskiwanie

**Punkt checklisty:** E5

**Severity:** HIGH

**Status dowodu:** CONFIRMED

## Co zaobserwowałem

Nod `Fetch Meta Ads` zaczyna się w `marketing-kpi-refresh-synthetic.json:38`, a jego obiekt `HTTP options` jest pusty w linii 47. Nie ma timeoutu na poziomie noda. W runie `MK-20260730-0745`, zaczynającym się w `marketing-report-runs-7d.json:49667`, wywołanie trwa 1855 sekund w linii 49690, a cały run pozostaje otwarty 1900 sekund. Dwa późniejsze zaplanowane starty zaczynają się przed jego zamknięciem.

## Dlaczego to ma znaczenie

Zawieszony dostawca zajmuje worker dłużej niż dwa interwały i dokłada się do F-03. Godzinny timeout workflow nie jest użyteczną kontrolą HTTP.

## Naprawa

Ustawić zmierzony timeout HTTP krótszy niż 15-minutowy interwał, skierować timeout do stanu held i dodawać retry dopiero po wdrożeniu ochrony przed overlapem i duplikacją.

## Jak zweryfikować naprawę

Opóźnić endpoint sandboxowy poza wybrany timeout. Nod HTTP ma przerwać pracę na limicie, żaden snapshot nie może zostać oznaczony jako kompletny, a kolejny zaplanowany run nie może konkurować z zakończonym timeoutem.

**Owner:** opiekun workflow | **Nakład:** S, poniżej 1 godziny

## 4. TOP-3 naprawy

**Dlaczego te trzy:** F-01 i F-02 zatrzymują nieodwracalne akcje. F-04 zatrzymuje decyzje oparte na raporcie tylko wyglądającym na kompletny.

### 1. Przenieść walidację webhooka przed wszystkie side effecty, F-01

**Dlaczego teraz:** Zły wynik występuje w nazwanym runie.

**Co robi klient:** Umieszcza validator za webhookiem i dopuszcza side effecty tylko na prawidłowej gałęzi.

**Jak zweryfikować:** Nieprawidłowy fixture wywołuje zero side effectów, prawidłowy wywołuje dokładnie jedno potwierdzenie.

**Owner:** opiekun workflow | **Nakład:** S | **Zależność:** brak

### 2. Dodać idempotencję realizacji i zawęzić retry, F-02

**Dlaczego teraz:** Jeden retry utworzył dwa rekordy docelowe.

**Co robi klient:** Przechowuje stabilny klucz, używa idempotencji dostawcy, wyszukuje przed utworzeniem, dodaje backoff i zatrzymuje po ostatnim błędzie.

**Jak zweryfikować:** Timeout po przyjęciu nadal pozostawia jeden rekord dostawcy po retry i replay.

**Owner:** opiekun workflow i właściciel API | **Nakład:** M | **Zależność:** brak

### 3. Blokować niepełne raporty dzienne, F-04

**Dlaczego teraz:** Dwa z siedmiu raportów pominęły źródło.

**Co robi klient:** Zatrzymuje błędy Google Ads i wymaga obu źródeł przed wysyłką.

**Jak zweryfikować:** Fixture z jednym źródłem nie wysyła normalnego raportu, fixture z dwoma źródłami wysyła jeden.

**Owner:** właściciel marketing operations | **Nakład:** M | **Zależność:** brak

## 5. Plan utwardzenia na 30 dni

| Okno                | Działanie                                                                                                                                      | Wynik, który ma istnieć później                                                                              |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| Pierwsze 24 godziny | Wdrożyć F-01. Tymczasowo wyłączyć automatyczne retry realizacji do czasu przejścia testu F-02. Zachować eksport dowodów z siedmiu dni.         | Nieprawidłowy webhook wywołuje zero side effectów, a drugi rekord realizacji nie może powstać automatycznie. |
| Dni 1 do 7          | Dokończyć F-02 i F-04. Dodać timeout HTTP z F-05, następnie lock współbieżności i uporządkowany upsert z F-03.                                 | Zapisanych pięć testów sandboxowych sprawdzających wynik biznesowy, a nie tylko zielone nody.                |
| Dni 8 do 14         | Dodać zewnętrzny freshness canary, outcome canary ze znanym wejściem i kontrolę hasha konfiguracji. Przypisać alerty do nazwanego właściciela. | Zatrzymany workflow, niepełny raport i celowa zmiana configu wywołują trzy różne, potwierdzone alerty.       |
| Dni 15 do 21        | Zmierzyć koszt platformy i API na zamówienie, odświeżenie KPI i poprawny raport dzienny. Ustawić limity runu i dnia.                           | Baseline i limity istnieją w konfiguracji, a okno pomiaru jest zapisane.                                     |
| Dni 22 do 30        | Odtworzyć oba workflow z eksportów JSON w środowisku testowym. Powtórzyć pięć testów akceptacyjnych i opisać odpowiedzialność za replay.       | Datowana notatka z odtworzenia, pięć zaliczonych testów, owner kolejki held i pisemna instrukcja replay.     |

## 6. Ograniczenia i luki danych

To jest raport-próbka oparty na celowo syntetycznych materiałach. Nie użyto eksportu klienta, systemu produkcyjnego, żywego endpointu ani prawdziwych danych osobowych. Wyniki po stronie dostawcy, takie jak zduplikowane identyfikatory docelowe, są jawnymi polami dowodowymi w syntetycznej historii, a nie obserwacjami z prawdziwego konta dostawcy.

Eksporty nie pokazują ustawień administracyjnych platformy, queue mode, concurrency workerów, walidacji na bramie przed webhookiem, dostarczania alertów, stanu backupu ani deduplikacji po stronie dostawców. Te kontrole mogą obniżyć albo podwyższyć ryzyko, dlatego obszary D i E nie dostały oceny PASS. Nie dostarczono billingu, więc raport nie szacuje kosztu.

Analizę automatyczną potraktowano wyłącznie jako kolejkę kandydatów. Jej agregat podał error rate 0.0 dla historii marketingowej, ponieważ status **warning** nie jest rozpoznawany jako błąd, mimo że dane nodów zawierają 3 błędy Google Ads. Reguła overlap oparta na p95 również pominęła 5 rzeczywistych nakładających się startów, bo wolne runy były rzadkie. Oba wnioski skorygowano przez bezpośredni odczyt statusów, timestampów, prób nodów i pół wyniku biznesowego. Ogólny kandydat freshness canary został odrzucony, ponieważ detektor generuje go na podstawie niestandardowego ustawienia, a nie dowodu dotyczącego zewnętrznego monitoringu.

Nic nie zostało uruchomione, powtórzone, edytowane ani wdrożone. Powyższe testy akceptacyjne są instrukcjami dla właściciela sandboxa, a nie twierdzeniem, że naprawy zostały zastosowane.

## Aneks A. Metoda

Audyt używał stałej checklisty 25 punktów w obszarach triggerów i harmonogramu, obsługi błędów i odzyskiwania, cichego driftu i jakości wyniku, kosztu i pętli oraz obserwowalności i backupu. Start, techniczne zakończenie i poprawny wynik biznesowy sprawdzano osobno. Toolkit sparsował graf i zagregował czasy runów, ale żaden wynik detektora nie został findingiem bez bezpośredniego potwierdzenia w surowym workflow albo nazwanym wykonaniu. Severity wynika ze skutku, prawdopodobieństwa i wykrywalności. Każde ustalenie CONFIRMED zawiera odtwarzalną lokalizację źródła oraz test akceptacyjny sprawdzający wynik biznesowy.

## Aneks B. Indeks dowodów

| Ref   | Artefakt                                                                                 | Co udowadnia                                                        |
|-------|------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| EV-01 | order-sync-v7-synthetic.json:16,23,44,59,247                                             | Utworzenie webhooka i potwierdzenie przed walidacją podpisu         |
| EV-02 | order-sync-runs-7d.json:1962,2028                                                        | Run z błędnym podpisem zapisał jedno potwierdzenie                  |
| EV-03 | order-sync-v7-synthetic.json:103,110,118                                                 | Trzy nieudane próby realizacji, brak klucza idempotencji i timeoutu |
| EV-04 | order-sync-runs-7d.json:1578,1628,1704                                                   | Trzy próby po przyjęciu i dwa rekordy docelowe                      |
| EV-05 | marketing-kpi-refresh-synthetic.json:16,172                                              | Hasło nie zgadło co 15 minut                                        |
| EV-06 | marketing-report-runs-7d.json:49764,49856,49861,49953                                    | Próba zapisania nowego                                              |
| EV-07 | marketing-kpi-refresh-synthetic.json:52,56,175,179,180                                   | Próba kontynuować do snapshotu i raportu                            |
| EV-08 | marketing-report-runs-7d.json:20990,21095,21098,58300,58405,58408                        | Próba zapisania jako udane                                          |
| EV-09 | marketing-kpi-refresh-synthetic.json:138,47<br>marketing-report-runs-7d.json:49667,49690 | Próba wywołania HTTP i wywołanie i trwające 1855 sekund             |

Prepared by SmartFlow - Independent x402 Observatory