



Projektowanie podprocesów w nAxiom

Wersja nAxiom: 1.13.2.0

Spis treści

1. Kluczowe informacje.....	2
2. Przykład biznesowy	3
3. Konfiguracja procesu głównego	3
4. Przykładowe rozwiązania projektowe.....	3
4.1. Tabela pomocnicza	7
4.2. Warunek synchronizacji.....	11
5. Podprocesy automatyczne	13

1. Kluczowe informacje

- Procesy biznesowe modelowane w nAxiom można zagnieżdżać. Proces nadrzędny jest wtedy nazywany *procesem głównym*, a proces podrzędny — *podprocesem*. Podprocesy mogą być inicjowane na dwa sposoby: synchronicznie i asynchronicznie.
- Blok podprocesu synchronicznego ma status i może mieć tyle wyjść, ile jest statusów dokumentu w podprocesie.
- Po zainicjowaniu podprocesu proces główny przyjmuje status bloku podprocesu i oczekuje aż przetwarzanie w podprocesie osiągnie żądany status.
- Po osiągnięciu statusu dokumentu w podprocesie proces główny wykonuje przejście przypisane do tego statusu.
- Blok podprocesu asynchronicznego nie ma statusu i ma jedno wyjście kontynuacji procesu głównego oraz może mieć tyle wyjść pomocniczych, ile jest statusów dokumentu w podprocesie.
- Po zainicjowaniu podprocesu proces główny wykonuje przejście do synchronizatora lub (przez blok decyzyjny) do dowolnego innego bloku. Aby zsynchronizować przetwarzanie w procesie głównym i podprocesie, wymagane jest zastosowanie bloku synchronizatora.
- Blok synchronizatora: określa moment (warunek synchronizacji) i kierunek przejścia procesu głównego w zależności od wyniku przetwarzania w podprocesie.
- Do bloku synchronizatora należy doprowadzić proces główny oraz wyjścia pomocnicze z jednego lub kilku bloków podprocesów asynchronicznych.

- Warunek synchronizacji: właściwość bloku synchronizatora, wyrażenie SQL zwracające 0 lub 1, zależnie od stanu przetwarzania w podprocesie (podprocesach).
- W podprocesach, które kończą się w momencie utworzenia dokumentu (*Start -> Utworzony*), wymagane jest umieszczenie po bloku startowym bloku oczekiwania (0 s), który zapewni prawidłowe przekazanie sterowania do procesu głównego. Takiego bloku należy również użyć przed blokiem podprocesu w przypadku wywoływania podprocesów w pętli; patrz osobny artykuł.

2. Przykład biznesowy

Wdrożenie nowego pracownika w firmie (*onboarding*) obejmuje kilka zadań wykonywanych zarówno przez tego pracownika, jak i pracowników odpowiednich działów (HR, IT itp.).

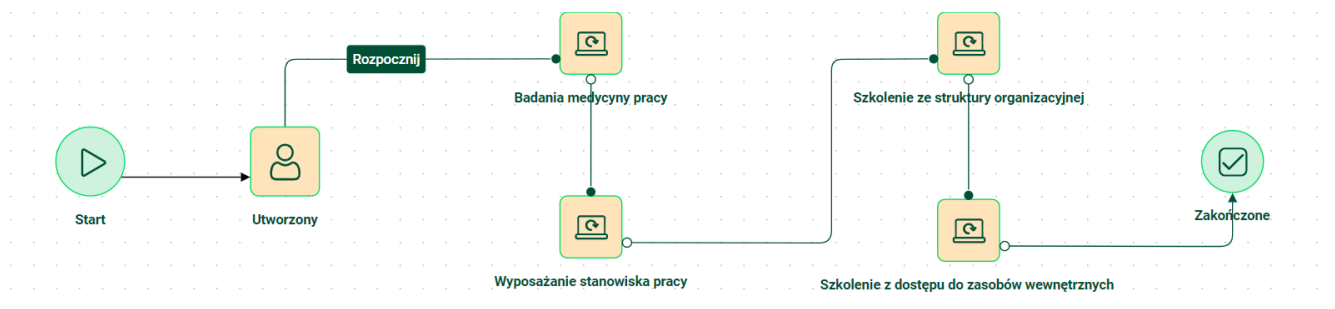


Proces onboardingu pracownika — główne etapy

Proces onboardingu to proces główny. Po jego zakończeniu pracownik będzie miał aktualne badania okresowe medycyny pracy, wypożyczenie stanowiska pracy, wiedzę na temat struktury organizacyjnej firmy oraz niezbędne dostępy do systemów informatycznych.

3. Konfiguracja procesu głównego

Etapy onboardingu mogą być realizowane sekwencyjnie w zadanej kolejności, jak na diagramie poniżej.

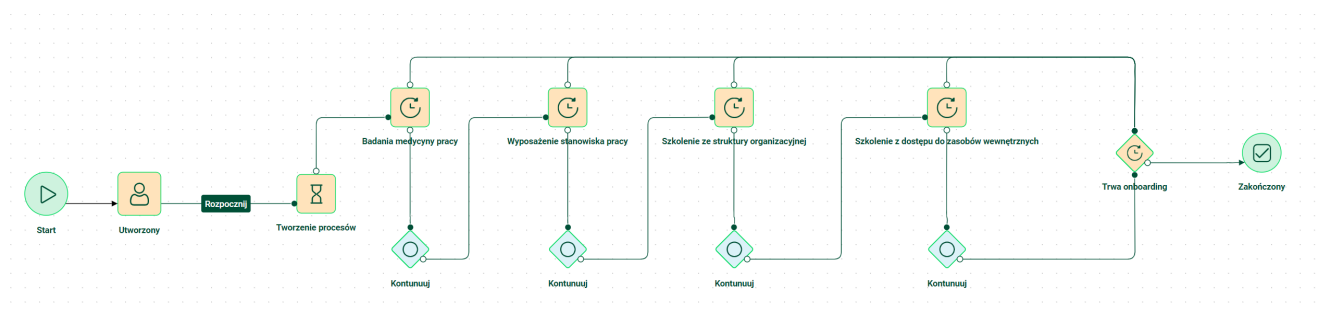


Podprocesy inicjowane synchronicznie

Pracownik działu HR inicjuje proces główny i pierwszy podproces. W podprocesie uprawniony pracownik wykonuje zatwierdzenie, które kończy podproces. Przejście do kolejnego etapu wymaga zakończenia etapu poprzedniego. Jeśli podproces osiągnie status przypisany do przejścia w procesie głównym, proces główny przejdzie do kolejnego bloku. Z bloku podprocesu można poprowadzić kilka przejść dla różnych statusów podprocesu.

W opisanym przypadku podprocesy są inicjowane synchronicznie. Proces główny inicjuje podproces, czeka w statusie przypisanym do tego podprocesu, a następnie przechodzi do kolejnego statusu reprezentowanego przez blok na diagramie procesów, gdy przetwarzanie w podprocesie „zakończy się” (z perspektywy procesu głównego).

Ten proces można również zaprojektować w taki sposób, aby kolejne etapy były realizowane niezależnie od siebie, jak na diagramie poniżej.



Podprocesy inicjowane asynchronicznie

W tym przypadku kolejne etapy są inicjowane asynchronicznie. Po kliknięciu **Rozpocznij** proces główny zainicjuje podprocesy kolejno, nie czekając na ich dalsze przetwarzanie, a po wywołaniu ostatniego podprocesu przejdzie do bloku synchronizatora i przyjmie jego status.

Asynchroniczne inicjowanie podprocesów wymaga zastosowania bloku podprocesu asynchronicznego. Ten blok nie ma statusu, ponieważ bezpośrednio po wywołaniu podprocesu

proces główny wykonuje przejście z zaznaczoną właściwością `Ścieżka kontynuacji procesu głównego`. Blok podprocesu może mieć tylko jedno takie przejście. W tym bloku jest jeszcze drugie wyjście — pomocnicze — które przekazuje informacje o osiągnięciu określonego statusu w podprocesie do bloku synchronizatora. Z tego wyjścia można poprowadzić tyle przejść, ile jest statusów w podprocesie.

W tym przypadku diagram procesu zawiera także dodatkowe elementy:

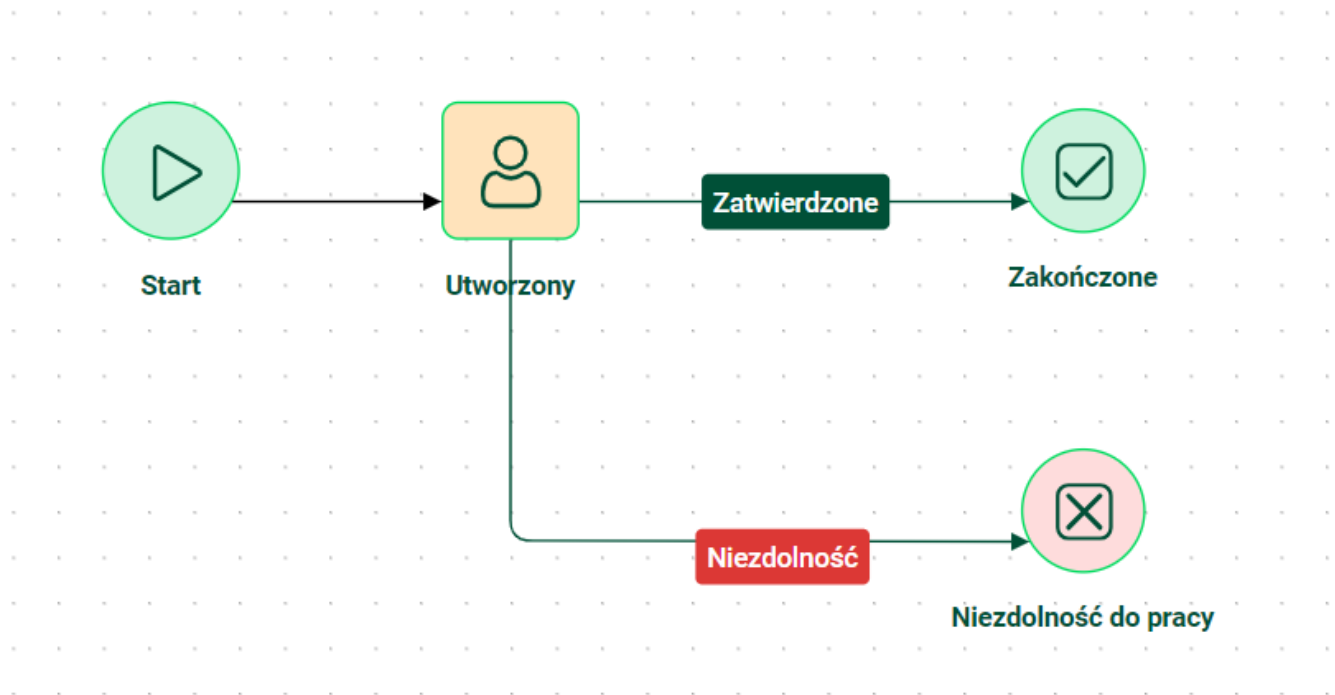
- Blok oczekiwania (0 s): może być wymagany w celu zamknięcie transakcji procesu głównego, co powoduje wyłączenie blokady ekranu (wirujący wskaźnik operacji w toku); dotyczy szczególnie sytuacji, kiedy blok podprocesu jest wywoływany w pętli, co w przypadku bardziej złożonych operacji może powodować przekroczenie limitu czasu otwartej transakcji (10 min). Wymagane jest ręczne odświeżenie formularza w celu aktualizacji statusu zgodnie z przejściem procesu głównego po zainicjowaniu podprocesu (podprocesów) - w opisywanym przypadku będzie to status synchronizatora.
- Blok decyzyjny na wyjściu kontynuacji procesu głównego z bloku podprocesu: wymagany ze względów technicznych jako blok *proxy*; wyjście procesu głównego z bloku podprocesu można połączyć tylko z blokiem decyzyjnym lub blokiem synchronizatora.
- Blok synchronizatora: reprezentuje etap procesu głównego, w którym proces główny oczekuje na zakończenie zainicjowanych podprocesów (możliwa jest także sytuacja, kiedy to zakończone podprocesy oczekują na proces główny). Określany we właściwościach tego bloku warunek synchronizacji pozwala określić, kiedy proces główny może przejść do następnego kroku, zależnie od statusu przetwarzania w podprocesach. Warunek jest sprawdzany, kiedy proces główny osiągnie status synchronizatora i kiedy w połączonych podprocesach zostanie osiągnięty status, dla którego poprowadzono przejście z wyjścia pomocniczego. Dodatkowo w synchronizatorze można wybrać kierunek przejścia procesu głównego.

4. Przykładowe rozwiązania projektowe

W przedstawionym powyżej diagramie procesu onboardingu z blokami podprocesów asynchronicznych scenariusz zakłada, że proces główny zakończy się, po zakończeniu

wszystkich podprocesów. Odpowiada za to warunek synchronizacji skonfigurowany w bloku synchronizatora. Osobny warunek odpowiada za wybór kierunku przejścia procesu głównego z synchronizatora. Połączenie tych dwóch warunków pozwala realizować bardziej złożone procesy biznesowe.

Jeśli na przykład podczas badań lekarskich zostanie stwierdzona niezdolność do pracy, proces główny może przejść do innego statusu, nie czekając na zakończenie przetwarzania w innych podprocesach.



Uproszczony diagram procesu badań lekarskich

W takim przypadku konieczne jest połączenie bloku podprocesu z synchronizatorem dodatkowym przejściem dla statusu *Niezdolność do pracy*. Dodatkowo w warunku synchronizacji należy dodać wyrażenie, które zwróci wartość 1, gdy dokument podprocesu badań zmieni status na *Niezdolność do pracy* oraz podobne wyrażenie w warunku wybierającym kierunek przejścia.

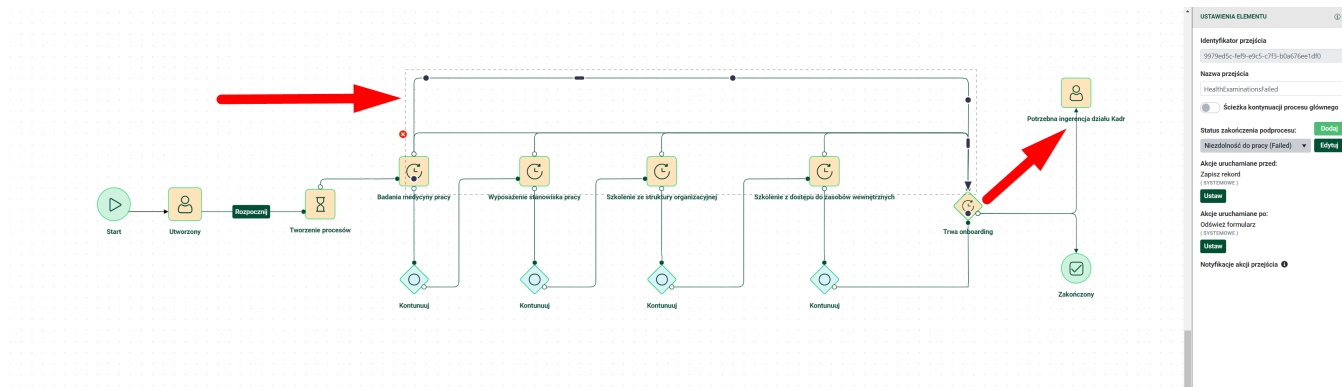


Diagram procesu głównego z obsługą przypadku niezdolności do pracy

4.1. Tabela pomocnicza

Wyrażenia warunku synchronizacji oraz wyboru kierunku przejścia muszą się odwoływać do tabel dokumentów tworzonych w podprocesach. Aby ułatwić ich konstrukcję, można utworzyć tabelę pomocniczą, w której będą zapisywane dane potrzebne do tych wyrażeń.



Edycja modelu: OnboardingAsyncChecklist

model tabeli pomocniczej

[dbo].[OnboardingAsyncChecklist] 



[dbo].[HealthExaminations] 



[

Lista pól

Dokumenty biznesowe

Formularze

Listy

Powiązania

Moduły



Nazwa pola danych

Typ danych

Id

Liczby całkowite

OnboardingId

Liczby całkowite

HealthExaminationsStatus

Tekst nvarchar

WorkstationEquipmentStatus

Tekst nvarchar

CompanyOrganizationStructureWorkshopStatus

Tekst nvarchar

InternalResourcesAccessWorkshopStatus

Tekst nvarchar

SyncConditionStatus

Tekst nvarchar

Przykład tabeli pomocniczej

Dla czytelności przykładu tabela składa się z klucza głównego, identyfikatora procesu głównego oraz umownych flag podprocesów. Dodatkowo, umowna flaga niepowodzenia badań lekarskich pozwoli łatwo wybrać kierunek przejścia.

W kreatorach mapowania podprocesów należy przekazywać do dokumentu podprocesu identyfikator procesu głównego, który umożliwi powiązanie dokumentów podprocesu z dokumentem procesu głównego.

Mapowanie zmiennych w podprocesie

Uzupełnij kolumny podprocesu zmiennymi z procesu głównego lub wartościami statycznymi.

Dostępne zmienne procesu głównego:

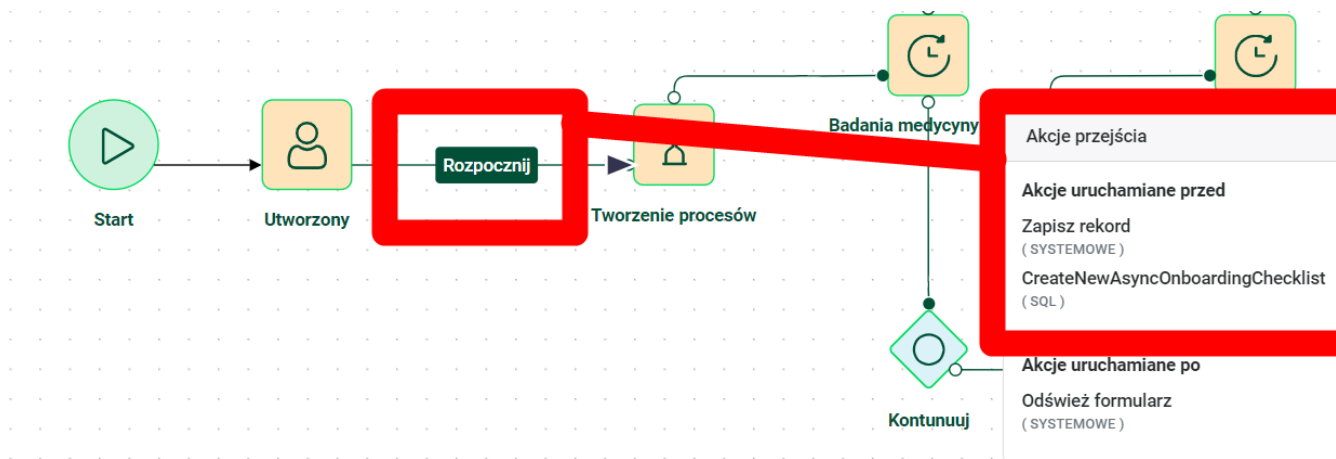
@ACLIId, @ArchiveStatus, @Blocked, @BlockUser, @Code, @CreateDate, @Id, @IsAnonimized, @LastEntryDate, @RowVersion, @Status, @Transition, @Type1, @Type2, @Type3, @UpdatedBy, @UpdatedDate, @UserId

Nazwa parametru	Wartość
Code	TXT ↔ 📄
IsAnonimized	TXT ↔ 📄
OnboardingId	TXT {@Id} ↔ 📄
Type1	TXT ↔ 📄
Type2	TXT ↔ 📄
Type3	TXT ↔ 📄

Zapisz
Anuluj

Przekazanie identyfikatora procesu głównego do podprocesu

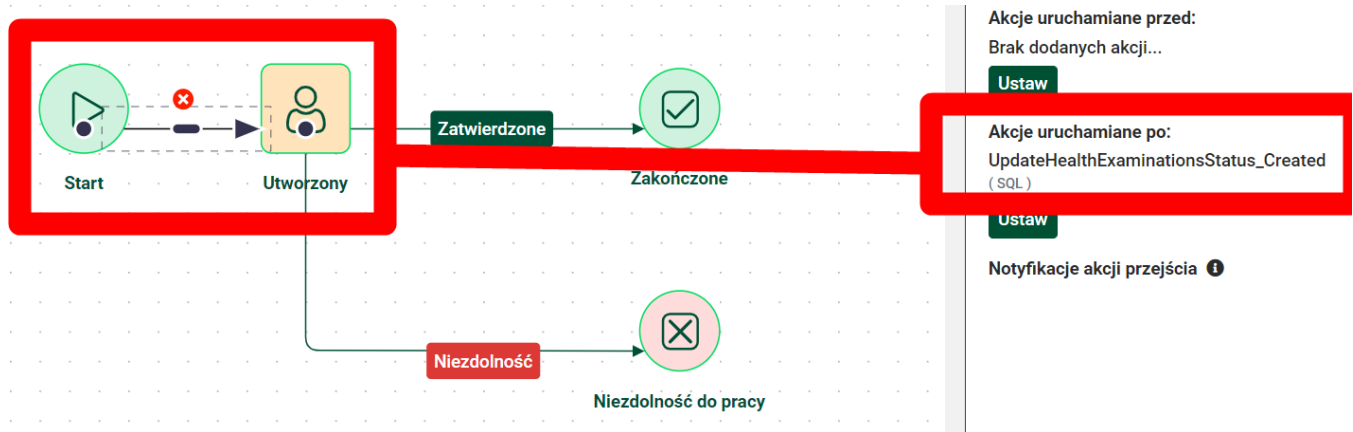
Na początku, w tabeli pomocniczej zostaje wstawiony nowy rekord z identyfikatorem procesu głównego. Służy do tego akcja SQL wykonywana przed zainicjowaniem pierwszego podprocesu.



Zapis danych do tabeli pomocniczej

```
INSERT INTO [dbo].[OnboardingAsyncChecklist] (OnboardingId) VALUES ({@Id})
```

Następnie, w każdym podprocesie należy skonfigurować akcję SQL, która zaktualizuje flagę podprocesu w odpowiednim rekordzie procesu głównego w tabeli pomocniczej.



Początkowa aktualizacja tabeli pomocniczej dla podprocesu badania lekarskiego

Przykładowe zapytania dla poszczególnych etapów podprocesu badań przedstawiono poniżej.

```
UPDATE [dbo].[OnboardingAsyncChecklist]
SET HealthExaminationsStatus = 'Waiting'
WHERE OnboardingId = {@OnboardingId}
```

```
UPDATE [dbo].[OnboardingAsyncChecklist]
SET HealthExaminationsStatus = 'Failed'
WHERE OnboardingId = {@OnboardingId}
```

```
UPDATE [dbo].[OnboardingAsyncChecklist]
SET HealthExaminationsStatus = 'Done'
WHERE OnboardingId = {@OnboardingId}
```

Z punktu widzenia rozpatrywanego scenariusza można ograniczyć się tylko do ustawienia flag *Failed* i *Done*. Analogiczne akcje SQL należy zdefiniować i przypisać także w pozostałych podprocesach.

4.2. Warunek synchronizacji

Przykładowy warunek synchronizacji badający jeden przypadek brzegowy (niezdolność do pracy) oraz ścieżkę podstawową (pozytywne ukończenie wszystkich etapów onboardingu) przedstawiono poniżej. Wykorzystano w nim sprawdzanie danych zapisanych w tabeli pomocniczej.

```
DECLARE @HealthExamStatus NVARCHAR(100)
DECLARE @WorkstationEquipmentStatus NVARCHAR(100)
DECLARE @OrgStrWorkshopStatus NVARCHAR(100)
DECLARE @ResourcesWorkshopStatus NVARCHAR(100)
DECLARE @thisId int = {@Id}

SET @HealthExamStatus = (
SELECT HealthExaminationsStatus FROM [dbo].[OnboardingAsyncChecklist]
WHERE OnboardingId = @thisId
)

SET @WorkstationEquipmentStatus = (
SELECT WorkstationEquipmentStatus FROM [dbo].[OnboardingAsyncChecklist]
WHERE OnboardingId = @thisId
)

SET @OrgStrWorkshopStatus = (
SELECT CompanyOrganizationStructureWorkshopStatus FROM [dbo].[OnboardingAsyncChecklist]
WHERE OnboardingId = @thisId
)

SET @ResourcesWorkshopStatus = (
SELECT InternalResourcesAccessWorkshopStatus FROM [dbo].[OnboardingAsyncChecklist]
WHERE OnboardingId = @thisId
)

IF @HealthExamStatus = 'Failed'
BEGIN
```

```

    UPDATE [dbo].[OnboardingAsyncChecklist]
    SET SyncConditionStatus = 'FailedOnHealthExaminations'
    WHERE OnboardingId = @thisId
    SELECT 1
END
ELSE IF @WorkstationEquipmentStatus = 'Done'
AND @HealthExamStatus = 'Done'
AND @OrgStrWorkshopStatus = 'Done'
AND @ResourcesWorkshopStatus = 'Done'
BEGIN
    UPDATE [dbo].[OnboardingAsyncChecklist]
    SET SyncConditionStatus = 'All done'
    WHERE OnboardingId = @thisId
    SELECT 1
END
ELSE
BEGIN
    SELECT 0
END

```

Kierunek przejścia (wybór wyjścia z synchronizatora) dla bieżącego dokumentu można określić na podstawie danych w kolumnie *SyncConditionStatus* w tabeli pomocniczej dla bieżącego dokumentu.

```

DECLARE @SyncConditionResult nvarchar(100)

SET @SyncConditionResult = (
    SELECT SyncConditionStatus FROM [dbo].[OnboardingAsyncChecklist]
    WHERE OnboardingId = {@Id}
)

IF @SyncConditionResult = 'FailedOnHealthExaminations'
    SELECT 'NeedAssist';
ELSE
    SELECT 'Finished'

```

5. Podprocesy automatyczne

W przypadku, gdy podproces wykonuje się automatycznie, na przykład obejmuje wyłącznie utworzenie dokumentu (tylko bloki Start i Utworzony), wymagane jest zastosowanie bloku oczekiwania z konfiguracją 0 sekund, najlepiej bezpośrednio po bloku startowym. Jest to istotne z technicznego punktu widzenia, ponieważ pozwala zarejestrować rozpoczęcie i zakończenie podprocesu (dwa osobne zdarzenia). Taką konfigurację przedstawiono poniżej.

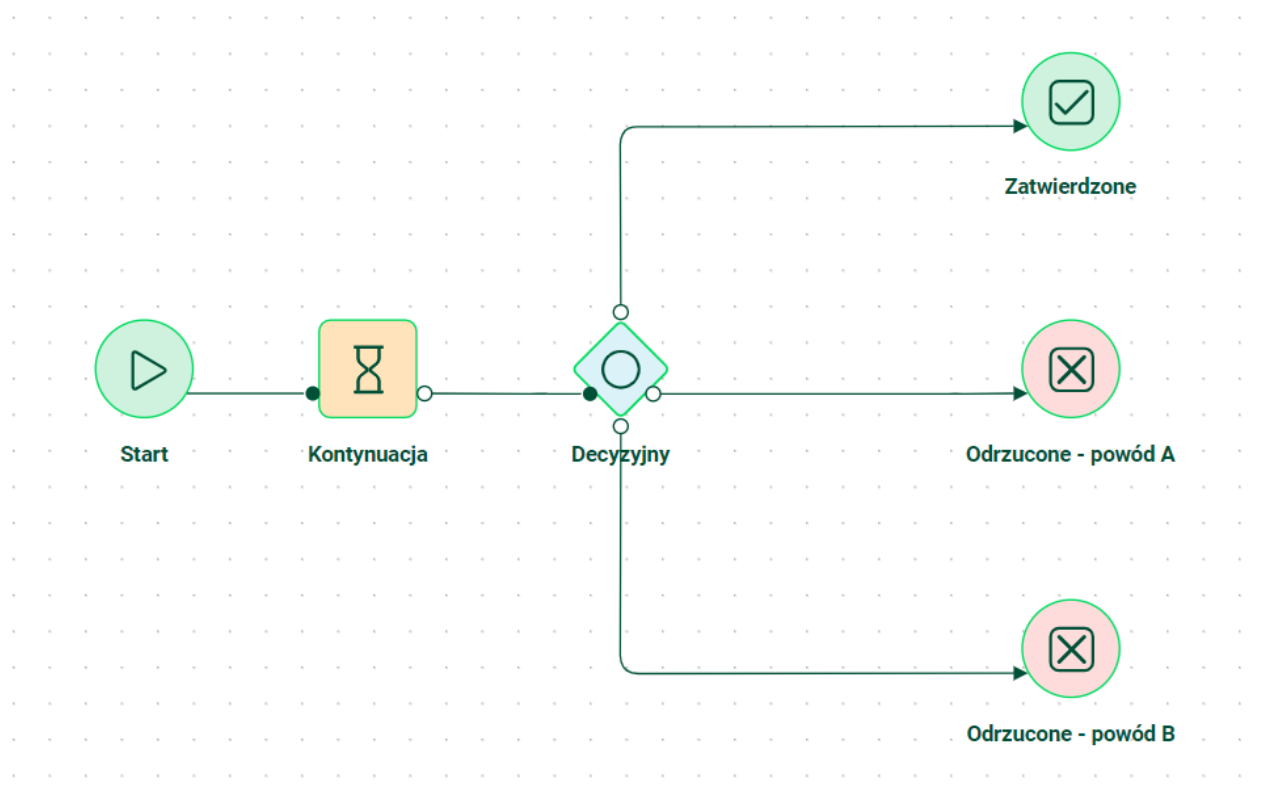


Diagram procesu automatycznego

Blok oczekiwania z konfiguracją 0 sekund ma też kluczowe znaczenie z punktu widzenia projektowania pętli podprocesów wywoływanych asynchronicznie z uwagi na właściwość zamykania transakcji bazy danych. Zasady projektowania takich pętli opisano w osobnym artykule.