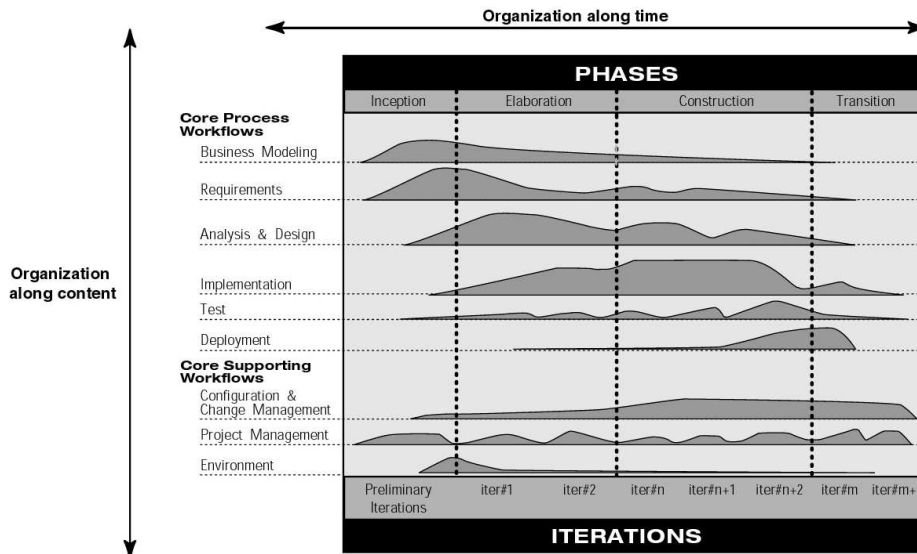


DIAGRAM PRZYPADKÓW UŻYCIA – USE CASE MODEL

1) Zastosowanie



- ▶ Jedną ze stosowanych metodologii prowadzenia projektów informatycznych (proces iteracyjnego wytwarzania oprogramowania wg IBM), jest **RUP (Rational Unified Process)**.
- ▶ Wg tej koncepcji, proces budowy oprogramowania obejmuje 4 fazy:
 - Faza rozpoczęcia (Inception phase)
 - Faza opracowywania (Elaboration phase)
 - Faza konstrukcji (Construction phase)
 - Faza przekazania systemu (Transition phase)
- ▶ RUP, podobnie jak inne techniki zarządzania projektami informatycznymi (np. Agile, eXtreme (XP), Scrum) stosuje „wytwarzanie sterowane przypadkami użycia (use-case) i scenariuszami”.
- ▶ Metodyki bazują na podstawowych elementach UML zgodnie z jego zasadami. Gdyby pominąć elementy metodyki zarządzania projektami niezwiązane z UML, to można zauważyć, że ogranicza się ona do określenia kolejności stosowania poszczególnych diagramów. W ten sposób UML stanowi uniwersalne narzędzie projektowania systemów komputerowych.



- ▶ Specyfikacja wymagań klienta, stworzenie wizji systemu, modelowanie biznesowe to zadania, które najczęściej operują na **scenariuszach przypadków użycia** (scenariuszach, scenariuszach użycia, scenariuszach użytkownika – user scenarios) opracowanych w formie tekstowej, stąd rozróżnienie na scenariusze i diagramy przypadków użycia.
- ▶ Dopiero z tych scenariuszy wyprowadzamy **diagram przypadków użycia**. Po przypadkach użycia zajmujemy się diagramem klas, a następnie bazując na nim budujemy diagramy sekwencji, a z nich możemy wygenerować diagram współpracy. Na koniec zostaje do wykonania diagram aktywności.
- ▶ Diagram przypadków użycia ilustruje **nazwy** przypadków użycia i aktorów oraz ich **związki**.
- ▶ Przypadek użycia reprezentuje **sekwencję działań** systemu, które przynoszą zauważalny rezultat w postaci **korzyści/użyteczności** dla aktora.
- ▶ Modelowanie za pomocą przypadków użycia (analiza) prowadzi do określenia **co** system ma robić, zamiast **jak** ma się to odbywać (projektowanie).

2) Konstrukcja

- ▶ Właściwie sformułowany Use Case powinien reprezentować **elementarny proces biznesowy** (EBP) – czyli zadanie wykonywane przez jedną osobę w jednym miejscu mające wartość biznesową oraz pozostawiające dane w stanie spójnym. Należy zwrócić uwagę, by nie definiować przypadków użycia na zbyt niskim poziomie (np. „zaloguj”, „drukuj”) – jako pojedynczych kroków czy pod-zadań w EBP. Główny przepływ zdarzeń powinien obejmować ok. 5-10 kroków i trwać około 5-10 minut w zależności od „rozdzielczości” systemu.
- ▶ Od powyższej zasady można stosować wyjątki tworząc „sub” use case’y, należy jednak pamiętać o łączeniu ich związkami z głównymi use case’ami.
- ▶ Identyfikacja aktorów, przypadków użycia i granic systemu opiera się na odpowiedziach na szereg pytań pomocniczych:
 - Jakie są granice systemu? Czy to tylko aplikacja komputerowa, czy oprogramowanie łącznie z hardware’em, a może również z osobą użytkownika?
 - Kim są podstawowi aktorzy, których cele są osiągnięte za pośrednictwem funkcjonalności systemu?

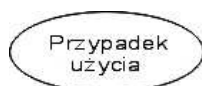
- Dla każdego aktora określ jego cele. Te cele powinny odpowiadać poziomowi szczegółowości przypadków użycia.
 - Określ przypadki użycia według listy celów, na każdy cel powinien przypadać jeden Use Case.
 - Pod-funkcje systemu (jak „zaloguj”) są dopuszczalne, jeśli ich obecność tłumaczy wielokrotna powtarzalność danego kroku w wielu przypadkach użycia.
 - Czy aktor musi czytać, modyfikować, tworzyć lub likwidować pewne informacje w systemie?
 - Czy aktor powinien być powiadomiony o zdarzeniach w systemie lub sam informować system? Co te zdarzenia reprezentują w kategoriach funkcji?
 - Jakich danych we/wy wymaga system? Skąd pochodzą i dokąd są kierowane?
- ▶ Aktor podstawowy (primary actor) to aktor, którego cele są spełnione; aktor pomocniczy (supporting actor) to aktor, który zapewnia usługi dla systemu.
 - ▶ Aktorem może być wszystko, co przedstawia pewne zachowanie. Aktorem może być człowiek, organizacja, program czy maszyna, a nawet omawiany system, jeśli wywołuje usługi innych systemów.
 - ▶ W określaniu aktorów podstawowych pomocne są odpowiedzi na pytania:
 - Czyje cele zostają spełnione?
 - Kto uruchamia i wyłącza system?
 - Kto administruje użytkownikami i bezpieczeństwem?
 - Czy istnieje monitoring pozwalający uruchomić system w razie awarii?
 - Jak dokonuje się aktualizacji systemu?
 - Kto ocenia sprawność systemu?
 - Jakimi urządzeniami zawiaduje system?
 - Z którymi systemami system ma współpracować (np. wymieniać dane)?
 - Kto jest zainteresowany rezultatami działania systemu?
 - ▶ **Przypadek użycia** – Use Case – składa się z kilku istotnych **sekcji**. Przede wszystkim ustalmy, że mówiąc o **przypadku użycia** mamy na myśli **pełny opis**, a nie sam **diagram** przypadków użycia, który ma na celu łatwą do ogarnięcia wizualizację przypadków użycia.

Nazwa:	Sprzedaj towar
Identyfikator:	UC1
Aktorzy:	Kasjer, Klient
Udziałowcy/Zainteresowani: (stakeholders/interests)	<ul style="list-style-type: none"> · Kasjer: oczekuje szybkiej i skutecznej transakcji, czas obsługi decyduje o jego zarobkach, braki w kasie potrącane są z wynagrodzenia. · Kierownik sprzedaży: oczekuje wzrostu prowizji ze sprzedaży sklepu. · Klient: oczekuje szybkiej obsługi i minimum wysiłku, żąda potwierdzenia (paragon, faktura) sprzedaży na wypadek reklamacji. · Sklep: oczekuje dokładnej rejestracji transakcji i zaspokojenia oczekiwań klientów. Żąda zapewnienia, że autoryzacja wpłat bezgotówkowych będzie rejestrowana. Oczekuje, że sprzedaż będzie możliwa, nawet jeśli niektóre komponenty systemu ulegną chwilowej awarii (np. terminal kart płatniczych). Wymaga automatycznej i bieżącej ewidencji produktów i

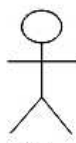
	<p>obrotów.</p> <ul style="list-style-type: none"> · Bank (system obsługi transakcji bezgotówkowych): żąda autoryzacji cyfrowej posiadacza karty w odpowiednim formacie i protokole. Oczekuje dokładnego rozliczenia transakcji ze sklepem. · Urząd podatkowy: wymaga pobrania podatku od każdego sprzedanego produktu.
Krótki opis:	Realizacja sprzedaży towaru
Warunki wstępne: (preconditions)	Klient musi posiadać wystarczające środki pieniężne oraz Kasjer musi posiadać odpowiedni towar. Kasjer musi pomyślnie zostać autoryzowany.
Warunki końcowe: (postconditions)	Sprzedaż zarejestrowana. Podatek obliczony. Rachunkowość i ewidencja towarów przeprowadzona. Prowizja aktualizowana. Wygenerowany paragon/faktura. Potwierdzenia autoryzacji karty zabezpieczone. Towar wydany.
Główny przepływ zdarzeń:	<ol style="list-style-type: none"> 1. Klient podchodzi do kasy zaopatrzonej w towar 2. Kasjer rozpoczyna nową sprzedaż 3. Kasjer wprowadza identyfikator towaru 4. System rejestruje kolejny towar i prezentuje jego nazwę, cenę i łączną cenę zakupów na wyświetlaczu <p>Kasjer powtarza kroki 3-4 aż do wyczerpania listy towarów</p> <ol style="list-style-type: none"> 5. System podlicza łączną kwotę i oblicza podatek 6. Kasjer przekazuje klientowi kwotę do zapłaty i pyta o formę płatności 7. Klient płaci a system realizuje płatność 8. System rejestruje pełną transakcję i przesyła informację do księgowości (dla rozliczeń i przeliczenia prowizji), ewidencji (dla aktualizacji inwentarza) 9. System drukuje potwierdzenie sprzedaży 10. Klient odchodzi z paragonem/fakturą oraz zakupionym towarem
Alternatywne przepływy zdarzeń:	<p>Np.</p> <ul style="list-style-type: none"> · · <p>5a. Klient zgłasza uprawnienie do rabatu:</p> <ol style="list-style-type: none"> 1. Kasjer wprowadza odliczenie rabatowe 2. Kasjer wprowadza identyfikację podstawy rabatu 3. System oblicza i wyświetla łączną kwotę do zapłaty po odliczeniu rabatu <ul style="list-style-type: none"> · · <p>7a. Klient płaci gotówką:</p> <ol style="list-style-type: none"> 1. Kasjer wprowadza kwotę pobraną 2. System oblicza resztę i otwiera kasę 3. Kasjer wpłaca gotówkę i wydaje resztę 4. System rejestruje płatność gotówką <p>7b. Klient płaci kartą:</p>

	<ol style="list-style-type: none"> 1. Klient podaje kartę i wprowadza PIN 2. System przesyła żądanie autoryzacji płatności do zewnętrznego systemu obsługi płatności i żąda potwierdzenia płatności. <ol style="list-style-type: none"> 2a. System otrzymuje informację o błędzie w transmisji: <ol style="list-style-type: none"> 1. System informuje o błędzie 2. Kasjer prosi o powtórzenie płatności lub płatność gotówką 3. System uzyskuje akceptację płatności z zewnętrznego systemu <ol style="list-style-type: none"> 3a. System otrzymuje odmowę płatności <ol style="list-style-type: none"> 1. System informuje Kasjera o odmowie 2. Kasjer prosi o płatność gotówką 4. System rejestruje płatność kartą i otrzymuje potwierdzenie. 5. System drukuje potwierdzenie płatności kartą <p style="text-align: center;">.</p> <p style="text-align: center;">.</p>
Specjalne wymagania:	<ul style="list-style-type: none"> • Ekran dotykowy. Teks ma być widoczny z 1 metra. • Płatności bezgotówkowe mają być realizowane w 30 sekund w 90% przypadków. • Menu ma być w kilku językach

- ▶ Powyższy przykład może być rozszerzony o dodatkowe sekcje, jak: zastosowane technologie, wymagane źródła danych, częstość występowania operacji, problemy, zagadnienia otwarte etc.
- ▶ Diagram przypadków użycia budujemy w oparciu o symbole:



Przypadek użycia prezentuje zbiór akcji które dostarczają aktorowi pożądanego zachowania



Aktor

Aktorem może być osoba lub inny system, którzy wchodzi w interakcje z systemem



Związek

Taki rodzaj związku będzie występował pomiędzy aktorem a konkretnym przypadkiem użycia



Stosujemy w przypadku zawierania się jednego przypadku w drugim, często w celu uproszczenia diagramu

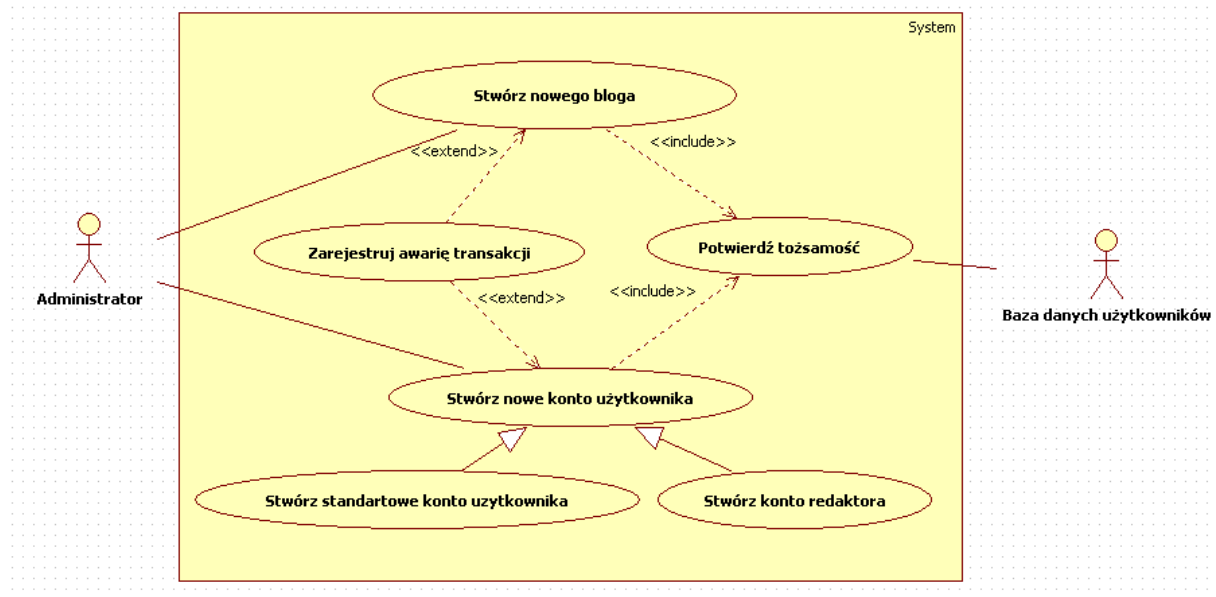


Stosujemy podczas rozszerzania przypadku użycia o inny, zwyczajowo gdy bazowy przypadek nie wystarcza



Symbol uogólnienia dotyczy sytuacji gdzie przypadek użycia dziedziczy wartości i zachowanie przodka, po czym uzupełnia je o swoje cechy

► Przykłady związków:



► Przykład:

