

# Analiza wymagań i dokumentacja

Ola Warzecha, 02.03.2018

# Po co o tym mówić?

- zbyt często na froncie panuje bałagan - programiści kodują na bazie własnego doświadczenia lub znanych patternów, a nie rzeczywistych oczekiwań klienta
- w projektach o dużym 'legacy' brak dokumentacji oznacza brak spójności nowych ficzerów oraz większe ryzyko błędów i pominięć
- klienci, UXowcy, graficy a nawet PMowie często nie mają świadomości, że analizy wymaga nie tylko backend, ale też frontend



# Przykład z życia - front

[link](#)



# Przykład z życia - backend

[link](#)



# Co stało się później?

- pocięcie widoku z uwagami - 2 tygodnie
- spięcie z backendem - 1 dzień
- odtworzenie procesu zakupowego, zależności i zidentyfikowanie wszystkich corner case'ów - trwa już 3 tydzień



# Co powinno się znaleźć w dokumentacji

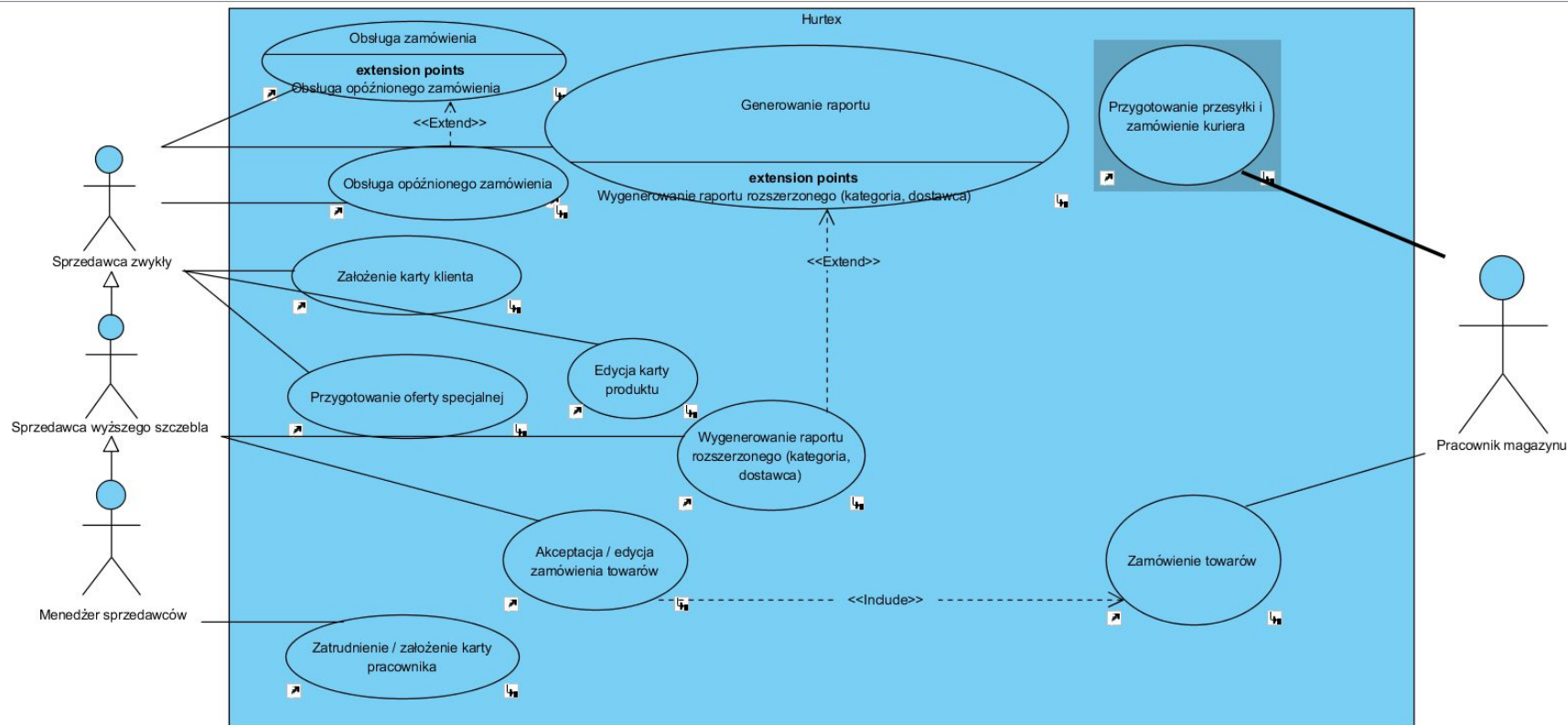
- ogólna struktura aplikacji, typy userów (aktorów) i zależności między nimi
- opis poszczególnych use case'ów z wyjątkami, corner case'ami, etc.
- makiety / projekty graficzne
- scenariusze testowe - pozytywne i negatywne
- szczegóły techniczne (komunikacja z API, itd.)



# Punkt wyjścia

Wobec rosnących obrotów, wzrastającej liczby klientów oraz poszerzającego się asortymentu sprzedaży – właściciele hurtowni artykułów spożywczych Hurtex postanowili wspomóc działanie firmy poprzez wprowadzenie systemu informatycznego. Poniżej przedstawiono najczęściej wykonywane w przedsiębiorstwie przez różnych pracowników czynności. Pracownik help-desku po odebraniu telefonu od klienta który zgłasza chęć zakupu produktu - przełącza rozmowę do sprzedawcy, który jest odpowiedzialny za klientów z określonego regionu a w przypadku gdy nie jest to możliwe ze sprzedawcą odpowiedzialnym za reprezentowany przez klienta obszar geograficzny. Po przekierowaniu połączenia – sprzedawca przyjmuje od klienta zamówienie tworząc kartę zamówienia na której umieszcza takie informacje jak kolejny numer zamówienia, dane klienta dla którego przygotowywane jest zamówienie, datę złożenia zamówienia, datę do kiedy zamówienie powinno być zrealizowane, informacje o ilości, cenie jednostkowej oraz ewentualnie udzielonym rabacie dla zamówionego asortymentu. Praktyka pokazuje że bardzo często w trakcie procesu przyjmowania zamówienia pomiędzy klientem a sprzedawcą prowadzone są negocjacje co do wysokości rabatu dla każdego zamawianego produktu (co oczywiście uzależnione jest od zamawianej ilości, dotychczasowej historii kontaktów z klientem etc.) Oczywiście jeśli się okaże iż klient zgłasza się do nas po raz pierwszy – przed przygotowaniem karty zamówienia przygotowywana jest karta klienta zawierające takie informacje jak nazwa firmy, dane osoby kontaktowej w firmie: np. szef działu zamówień, czy też osoba która się z nami skontaktowała etc. oraz dane teled adresowe Sprzedawcy na koniec każdego dnia/tygodnia/miesiąca/kwartału/roku generują raport (ilościowy i jakościowy) własnej sprzedaży za określony dzień/tydzień/miesiąc/kwartał/rok. Sprzedawcy wyższego szczebla generują raporty (ilościowe i wartościowe) sprzedaży produktów z danej kategorii (w wybranym okresie) (mięsa, warzywa, owoce etc.) Celem określenia najbardziej- oraz najmniej- zyskownych kategorii produktu Sprzedawcy wyższego szczebla generują raporty (ilościowe i wartościowe) sprzedaży (w wybranym okresie) produktów dostarczanych przez konkretnego sprzedawcę – aby ewentualnie negocjować lepsze ceny zakupu towarów Sprzedawcy (dowolnego szczebla) generują raporty przedstawiające sprzedaż konkretnego produktu (ilość, wartość sprzedaży przed rabatem, wartość sprzedaży po rabacie). Przy czym raport musi uwzględniać wartość sprzedaży z dnia sprzedaży a nie z dnia generowania raportu Sprzedawca w przypadku wątpliwości co do ceny czy parametrów produktu wysyła do dostawcy zapytanie o tenże produkt ewentualnie sprawdza te dane na stronie WWW dostawcy (i w przypadku zmian takich parametrów jak cena jednostkowa, ilość-rozrachunkowa – modyfikuje te dane w karcie produktu)...

# Zdefiniowanie ról i use case'ów





# Przykładowy use case - generowanie raportu

[link](#)



# Scenariusze testowe

[link](#)

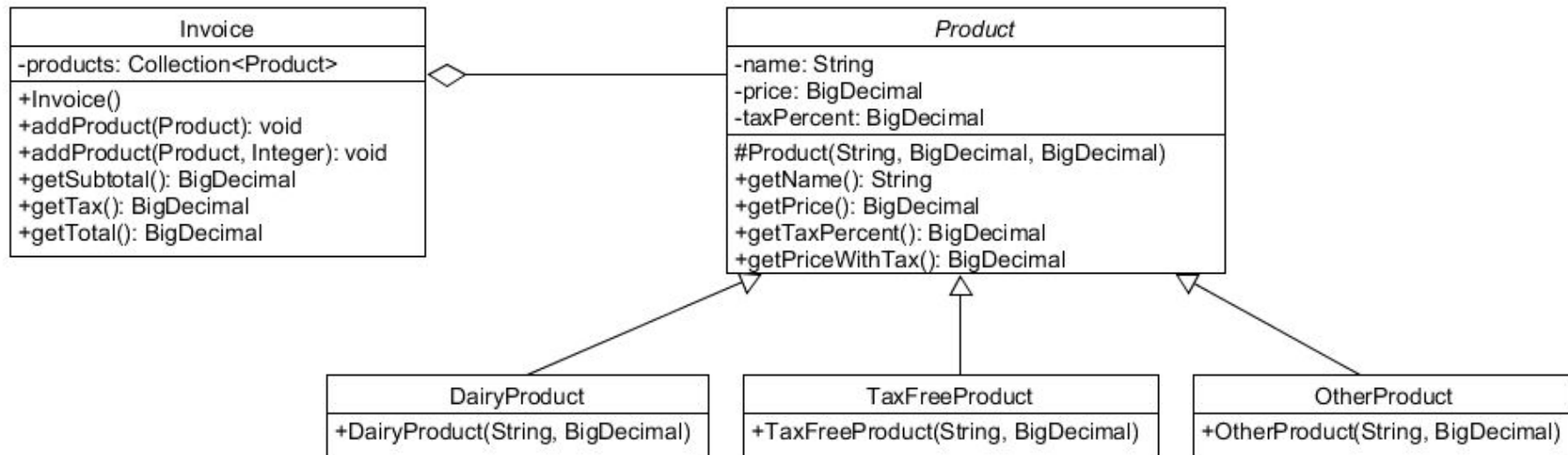


# Kompletne dokumenty

[link](#)



# Co dalej? Analiza obiektowa, diagramy klas...



# Co my możemy z tym zrobić?

- zadawać dużo pytań i mieć odpowiedzi na piśmie - np. w Jirze. Podstawowe pytania:
  - jakie mamy typy użytkowników i ich uprawnienia
  - jakie są scenariusze które uwzględniamy (include lub extend)
  - jak przebiega scenariusz, który programujemy i jakie ma wyjątki
  - API, przykładowe odpowiedzi na różnych przypadków
- wymagać dokumentacji
- uzupełniać analizę wymagań i dokumentację samemu w trakcie pracy (choćby sam opis UC + komentarze w kodzie), jeśli to my decydujemy jak dana funkcjonalność działa

