
ZARZĄDZANIE CYKLEM ŻYCIA APLIKACJI
INŻYNIERIA WYMAGAŃ

WPROWADZENIE

INŻYNIERIA WYMAGAŃ

PRZYKŁAD – SYSTEM REZERWACJI

Wyobraźmy sobie, że przychodzi do nas klient, który prowadzi:

- Przychodnię lekarską
- Restaurację
- Escape room

Każdy z nich chciałby, żebyśmy stworzyli system za pomocą którego ich klienci mogliby dokonać rezerwacji przez internet.

Co tak naprawdę powinniśmy zrobić?

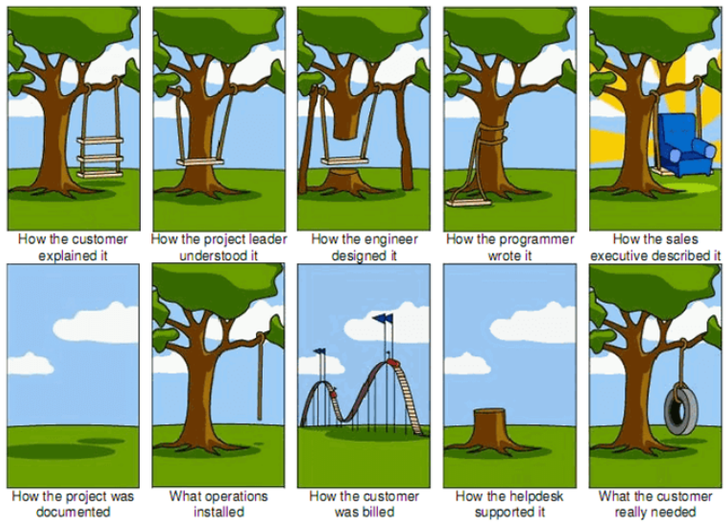
PRZYKŁAD – SYSTEM REZERWACJI

Samo powiedzenie „system rezerwacji online” to za mało.

Każdy z klientów jest z innej branży i ma inne oczekiwania względem takiego systemu, który ma być dostosowany do potrzeb jego i jego użytkowników.

Zarówno klient jak i my mamy pewną wizję (prawdopodobnie zupełnie różną!) tego systemu w głowie.

PERSPEKTYWY A KOMUNIKACJA



Efekt końcowy może być skutkiem:

- różnych perspektyw
- różnych dążeń, interesów
- różnych doświadczeń życiowych
- nieprecyzyjnej komunikacji
- niezagwarantowania, nieupwenienie się, że wszystkie strony kontraktu rozumieją te same pojęcia, wymagania i cele w ten sam sposób

PRZYKŁAD – SYSTEM REZERWACJI

Jak uniknąć nieporozumień?

Określając i dokumentując wymagania.

PRZYKŁAD – SYSTEM REZERWACJI

Przykładowe wymagania dla systemu rezerwacji w restauracji

- Rejestracja nowego klienta
- Możliwość określenia godziny rozpoczęcia.
- Możliwość określenie godziny zakończenia.
- Możliwość wykonania rezerwacji za pomocą urządzenia mobilnego.
- Możliwość wyboru stolika
- System wysyła potwierdzenie rezerwacji SMSem
- SMS wysyłany jest do 5 minut od momentu rezerwacji.
- Maksymalnie 2 godziny potrzebne na przeszkolenie pracownika, który będzie obsługiwał ten system.

PRZYKŁAD – SYSTEM REZERWACJI

Przykładowe wymagania dla systemu rezerwacji w Escape Room

- Możliwość wyboru jednego z trzech pokoi.
- Możliwość wyboru godziny wizyty.
- Możliwość wyboru liczby gości.
- Możliwość płatności online.
- Możliwość udostępnienia rezerwacji na Facebooku.
- Do 15 minut po wizycie system wysyła maila z prośbą o ocenę na 5 gwiazdek w jakimś serwisie oceniającym Escape Roomy.

PRZYKŁAD – SYSTEM REZERWACJI

Przykładowe wymagania dla systemu rezerwacji w przychodni lekarskiej

- Jakieś propozycje?

PRZYKŁAD – SYSTEM REZERWACJI

- Czy mamy pewność, że określiliśmy wszystko?
- Czy możemy te określone przez nas wymagania dodać jako załącznik do umowy, określić cenę i podpisać?
- Czy określiliśmy co się ma stać w Escape Roomie jak dwóch klientów zarezerwuje w tym samym czasie?
- Czy określiliśmy co się stanie, jeżeli w przychodni lekarskiej lekarz zachoruje i będziemy musieli odwołać(a może przełożyć?) wizyty pacjentów.
- Czym w ogóle jest wymaganie?

CZYM JEST WYMAGANIE?

Cecha, którą system musi mieć lub ograniczenie, które musi spełniać, żeby mógł być zaakceptowany przez klienta.

INŻYNIERIA WYMAGAŃ

Stawia sobie za cel określenie wymagań wobec konstruowanego systemu.

WYMAGANIA FUNKCJONALNE

Wszystkie funkcje systemu widziane od strony użytkownika oraz otoczenia zewnętrznego (np. systemów, z którymi się integrujemy).

WYMAGANIA POZAFUNKCJONALNE

Pozostałe wymagania określające jakość i granice, w których wymagania funkcjonalne mają być realizowane.

MODEL FURPS

Functionality - funkcjonalność, wymaganie funkcjonalne

Usability - użyteczność, łatwość użytkowania systemu

Reliability - niezawodność

Performance - wydajność

Security - bezpieczeństwo

URPS uznaje się za wymagania pozafunkcjonalne

FUNKCJONALNOŚĆ (F)

- Zdolność systemu do zaspokajania potrzeb użytkowników przy użyciu dostępnych funkcji
- Może być mierzona:
  - Liczbą funkcji istotnych z perspektywy użytkownika, które oferuje dany produkt, np. bankowość internetowa

UŻYTECZNOŚĆ (U)

- Oznacza łatwość użytkowania systemu
- Może być mierzona przez:
  - Czas szkolenia pracowników
  - Liczbę kontaktów z klientem
  - Liczbę sytuacji, w której konieczne jest skorzystanie z systemu pomocy.
  - Liczba błędów, które użytkownik popełnia przechodząc przez jakiś proces

NIEZAWODNOŚĆ (R)

- Szczególnie istotna w kontekście systemów pracujących w sposób ciągły
- Może być mierzona przez:
  - Średnią liczbę godzin pracy bez awarii
  - Maksymalną liczbę godzin w miesiącu, w których system może byc wyłączony

WYDAJNOŚĆ (P)

- Morze być mierzona przez:
  - Liczbę operacji na godzinę
  - Liczbę użytkowników, którzy korzystają z systemu jednocześnie
  - Szybkością ładowania strony
    - https://developers.google.com/speed/pagespeed/insights/

BEZPIECZEŃSTWO (S)

- Wymagania związane z tajnością danych, polityką prywatności, przechowywaniem danych osobowych.
- Może być określane poprzez:
  - Użycie algorytmu szyfrowania
  - Spełniane standardy bezpieczeństwa

Wymagania bezpieczeństwa mogą pochodzić z początkowych założeń, ale też mogą pojawić się w wyniku przeprowadzenia przez firmę audytującą skanu bezpieczeństwa, testów penetracyjnych.

// Przykład PEN testu

ŹRÓDŁA WYMAGAŃ

Greenfield engineering

- nowy projekt
- wymagania pochodzą od klientów i użytkowników.

Reengineering

- przeprojektowanie i ponowna implementacja systemu
- wymagania pochodzą ze starego systemu

WYMAGANIA FUNKCJONALNE

- Opisują funkcje systemu dostępne dla różnych użytkowników
- Podejścia do opisu wymagań funkcjonalnych:
  - „System powinien…”
  - Lista funkcji systemu
  - Przypadki Użycia (Use Cases)
  - User stories

Przedstawimy 3 podejścia. Pierwsze dwa są podejściami historycznymi, natomiast przypadki użycia i user stories są współcześnie powszechnie stosowanymi formami dokumentacji wymagań.

„SYSTEM POWINIEN…”

1. System powinien umożliwić wystawianie faktur
2. System powinien generować zestawienie miesięczne faktur
3. Faktura powinna zawierać conajmniej jedną pozycję
4. Faktura powinna być przypisana do najwyżej jednego klienta

i.t.d.

Zalety

- Łatwość spisywania

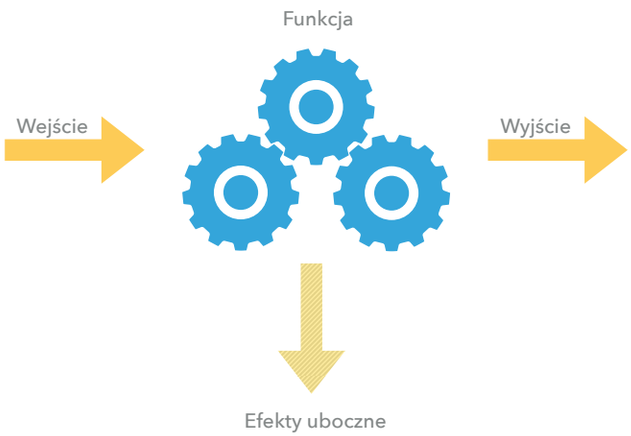
Wady

- Mała czytelność
- trudność w sprawdzaniu kompletności, spójności

Wymagania w stylu „System powinien...” były intensywnie wykorzystywane w latach 80-tych, 90- tych. W roku 1998 nawet zostały zaproponowane jako standard (IEEE 830-1998). Ich dużą zaletą jest łatwość spisywania.

Niestety obecne systemy stają się coraz bardziej złożone, a wymagania średniej wielkości systemu w tej postaci zajęłyby kilkaset stron. Tak wielka liczba luźnych zdań, niepowiązanych ze sobą sprawia trudności podczas sprawdzania jakości takiej specyfikacji.

FUNKCJE SYSTEMU



Wady

- Mała czytelność
- Trudne do zrozumienia

Innym podejściem jest opisywanie poszczególnych funkcji systemu. Analogicznie do funkcji matematycznych, każda funkcja systemu informatycznego ma swoje wejście, wyjście, efekty uboczne.

Przykładowo, rozpatrując funkcję wystawiania faktury, wejściem mogą być pozycje faktury, wyjściem wydruk faktury (lub wysłanie jej faksem), natomiast efektem ubocznym zapisanie tej faktury w rejestrze faktur.

Podobnie jak w przypadku wymagań typu „System powinien”, taka specyfikacja wymagań zawiera bardzo dużą liczbę małych funkcji, więc nadal są problemy ze zrozumieniem idei systemu i czytelnością.

INŻYNIERIA WYMAGAŃ

PRZYPADKI UŻYCIA (USE CASES)

Aktor	Sprzedawca	Rozszerzenia
Warunek początkowy	- Sprzedawca jest zalogowany do systemu - Sprzedawca chce wystawić fakturę	
Warunek końcowy	- Wygenerowana faktura w formie pliku PDF zostaje zapisana w repozytorium	
Główny scenariusz powodzenia	1. Sprzedawca rozpoczyna proces dodawania nowej faktury 2. Sprzedawca wprowadza pozycje faktury 3. Sprzedawca wprowadza dane firmy klienta (nazwa, adres, NIP) 4. System waliduje poprawność NIP 5. Sprzedawca zatwierdza poprawność wprowadzonych danych 6. System podlicza fakturę, nadaje jej numer, zapisuje w rejestrze i generuje fakturę do pliku PDF	3a. Dane firmy są niepoprawne 3a1. Sprzedawca anuluje proces generowania faktury 3a2. Powrót do punktu 3.

Zalety

- Łatwość spisywania
- Czytelność
- Łatwość zrozumienia i wyobrażenia sobie przyszłego systemu

Wady

- Dużo pisania

Trzecie podejście to przypadki użycia.

Poprzednie podejścia opisywały wymagania z perspektywy systemu - jak system powinien się zachować w określonej sytuacji. Przypadki użycia natomiast skupiają się na interakcji pomiędzy użytkownikiem, a systemem.

Dzięki takiemu podejściu zyskujemy na czytelności - przypadki użycia nie pokazują zbędnych szczegółów. Dużo łatwiej też jest klientowi wyobrazić sobie jak system będzie funkcjonował. Czyta pewną opowieść, która mówi o tym, w jaki sposób np. wystawić fakturę.

USER STORIES

User story

Jako sprzedawca, który dodaje nową fakturę, chcę mieć możliwość wprowadzenia najpierw numeru NIP klienta, aby wszystkie pola danych firmy zostały automatycznie uzupełnione, jeżeli dane klienta znajdują się w bazie.

Kryteria akceptacyjne

- w polu do wprowadzenia numeru NIP użytkownik może wprowadzić ciąg 10 cyfry z zakresu 0-9
- jeśli użytkownik wprowadzi mniej niż 10 dozwolonych znaków zostanie wyświetlony stosowny komunikat
- w polu nie można wprowadzić znaków spoza zdefiniowanego zakresu
- pole będzie walidowane „w locie”

Zalety

- Łatwość spisywania
- Czytelność
- Mnie sformalizowane, zdroworozsądkowe podejście

Wady (?)

- Nie dostarczają szczegółowej specyfikacji wymagania

User Story to najmniejsza jednostka pracy w zwinnym podejściu do wytwarzania oprogramowania.

User Story to kilka zdań w prostym języku, które opisują pożądany rezultat. Nie wchodzą w szczegóły. Wymagania są dodawane później, po uzgodnieniu przez zespół.

User Story nie jest funkcją jako taką, lecz celem wyrażonym z punktu widzenia użytkownika, do osiągnięcia którego dąży zespół.

OMÓWIENIE I ĆWICZENIA

PRZYPADKI UŻYCIA (USE CASES)

PRZYPADKI UŻYCIA (USE CASES)

PRZYPADEK UŻYCIA (UC) JEST OPISEM INTERAKCJI POMIĘDZY UŻYTKOWNIKIEM A SYSTEMEM INFORMATYCZNYM.

Jak już mówiliśmy - przypadek użycia jest opisem interakcji pomiędzy użytkownikiem, a systemem informatycznym.

Mogą one występować w różnych formach. Najprostsza to akapit tekstu pisany językiem naturalnym. Bardziej powszechna jest jednak postać strukturalna.
Każdy przypadek użycia w tej formie składa się z następujących elementów:
Nazwa - wyraża cel przypadku użycia (np. wystawianie faktury, zamówienie książki, dodanie wpisu na forum)

PRZYPADEK UŻYCIA (USE CASE)

Każdy przypadek użycia składa się z następujących elementów:

PRZYPADEK UŻYCIA (USE CASE)

Wystawienie faktury

- Nazwa

Nazwa - wyraża cel przypadku użycia (np. wystawianie faktury, zamówienie książki, dodanie wpisu na forum)

PRZYPADEK UŻYCIA (USE CASE)

UC1: Wystawienie faktury

- Nazwa
- Identyfikator

Identyfikator - unikalny w obrębie danej specyfikacji wymagań, przydaje się podczas dyskusji na temat wymagań - do odwoływania się do poszczególnych elementów. Dzięki nim można w łatwy sposób odwołać się do przypadku użycia, np. w UC1 w kroku 3 jest błąd.

PRZYPADEK UŻYCIA (USE CASE)

- Nazwa
- Identyfikator
- Aktor

UC1: Wystawienie faktury

Aktor	Sprzedawca	

nazwę głównego aktora - użytkownika, który gra główną rolę w danym przypadku użycia

PRZYPADEK UŻYCIA (USE CASE)

- Nazwa
- Identyfikator
- Aktor
- Warunek początkowy

UC1: Wystawienie faktury

Aktor	Sprzedawca	
Warunek początkowy	- Sprzedawca jest zalogowany do systemu - Sprzedawca chce wystawić fakturę	

Warunek początkowy - określa stan wyjściowy systemu, warunki konieczne do spełnienia, zanim scenariusz główny działania będzie mógł zostać zrealizowany

PRZYPADEK UŻYCIA (USE CASE)

- Nazwa
- Identyfikator
- Aktor
- Warunek początkowy
- Warunek końcowy

UC1: Wystawienie faktury

Aktor	Sprzedawca	
Warunek początkowy	- Sprzedawca jest zalogowany do systemu - Sprzedawca chce wystawić fakturę	
Warunek końcowy	- Wygenerowana faktura w formie pliku PDF zostaje zapisana w repozytorium	

Warunek końcowy, gwarancja potwierdzenia - określa stan wyjściowy systemu, pożądany rezultat

PRZYPADEK UŻYCIA (USE CASE)

- Nazwa
- Identyfikator
- Aktor
- Warunek początkowy
- Warunek końcowy
- Główny scenariusz

UC1: Wystawienie faktury

Aktor	Sprzedawca	
Warunek początkowy	- Sprzedawca jest zalogowany do systemu - Sprzedawca chce wystawić fakturę	
Warunek końcowy	- Wygenerowana faktura w formie pliku PDF zostaje zapisana w repozytorium	
Główny scenariusz powodzenia	1. Sprzedawca rozpoczyna proces dodawania nowej faktury 2. Sprzedawca wprowadza pozycje faktury 3. Sprzedawca wprowadza dane firmy klienta 4. Sprzedawca zatwierdza poprawność wprowadzonych danych 5. System podlicza fakturę, nadaje jej numer, zapisuje w rejestrze i generuje fakturę do pliku PDF	

Główny scenariusz - sekwencja kroków, która musi zostać wykonana do osiągnięcia celu przypadku użycia. Opisuje najczęstszy przebieg interakcji pomiędzy głównym aktorem, a systemem.

PRZYPADEK UŻYCIA (USE CASE)

- Nazwa
- Identyfikator
- Aktor
- Warunek początkowy
- Warunek końcowy
- Główny scenariusz
- Rozszerzenia

UC1: Wystawienie faktury

Aktor	Sprzedawca	Rozszerzenia
Warunek początkowy	- Sprzedawca jest zalogowany do systemu - Sprzedawca chce wystawić fakturę	
Warunek końcowy	- Wygenerowana faktura w formie pliku PDF zostaje zapisana w repozytorium	
Główny scenariusz powodzenia	1. Sprzedawca rozpoczyna proces dodawania nowej faktury 2. Sprzedawca wprowadza pozycje faktury 3. Sprzedawca wprowadza dane firmy klienta (nazwa, adres, NIP) 4. System waliduje poprawność NIP 5. Sprzedawca zatwierdza poprawność wprowadzonych danych 6. System podlicza fakturę, nadaje jej numer, zapisuje w rejestrze i generuje fakturę do pliku PDF	4a. Wprowadzony NIP jest niepoprawny 4a1. Sprzedawca anuluje proces generowania faktury 4a2. Powrót do punktu 3.

Nie zawsze główny scenariusz może zakończyć się powodzeniem, czy to z powodu błędu użytkownika czy z powodu niespełnienia jakichś warunków.

Np. Podczas wybierania gotówki z bankomatu główny scenariusz powodzenia zakłada, że użytkownik wprowadza poprawny PINu, że karta jest ważna, że na koncie znajdują się środki.

Sytuacje w których PIN okazuje się nie poprawny, karta straciła ważność, użytkownik nie posiada środków na swoim koncie są opisywane właśnie w Rozszerzeniach.

PRZYPADEK UŻYCIA (USE CASE)

- Atrybuty
  - Źródło wymagań
  - Priorytet
  - Status przypadku użycia

Źródło wymagań - tzn. skąd, od kogo dane wymaganie pochodzi, np. Anna Kowalska, Manager Produktu lub wynik analizy konkurencyjnego serwisu (odesłanie do dokumentacji)

Priorytet - jak ważny jest dane wymaganie w kontekście całości systemu, czy jest częścią MVP czy może być dostarczone w kolejnych fazach

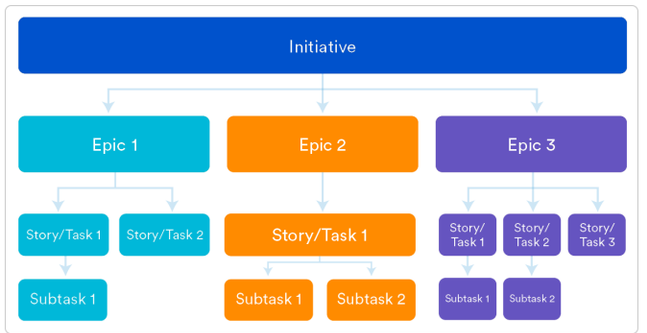
Status UC - W momencie opracowywania wymagań różne UC mogą posiadać różny status, np.
(1) w trakcie opracowywania przez zespół projektowy, (2) oczekujący na akceptację po stronie interesariuszy lub klienta, (3) zaakceptowany przez klienta - dobra praktyką jest aby tylko takie wymagania były przekazywane do dalszych prac, (4) - zdezaktualizowany

OMÓWIENIE I ĆWICZENIA

USER STORIES

CZYM JEST USER STORY?

- Najmniejsza jednostka pracy w zwinnych metodykach pracy
- Celem wyrażonym z punktu widzenia użytkownika
- Kilka zdań w prostym języku, które opisują pożądany rezultat
- Nie wchodzą w szczegóły - wymagania dodawane są później po uzgodnieniu przez zespół
- Elementy składowe większych jednostek takich jak Epic, Initiative



Te większe struktury zapewniają, że codzienna praca zespołu programistycznego przyczynia się do celów organizacyjnych wbudowanych w epiki i inicjatywy.

Inicjatywa: migracja/przeplatformowanie kilku serwisów kontentowych z jednej platformy na inną (np zmiana CMS)

Epic: migracja pojedynczego serwisu

Komponent: zbudowania modułu logowania dla nowej platformy

Story: Jako user, który zapomniał hasła logowania do serwisu, chcę mieć możliwość jego zmiany, tak aby ponownie uzyskać dostęp do panelu zalogowanego użytkownika.

INŻYNIERIA WYMAGAŃ

JAKIE KORZYŚCI DAJĄ USER STORIES?

- Skupiają uwagę na użytkowniku
- Wspierają współpracę
- Napędzają kreatywne rozwiązania
- Napędzają dynamikę zespołu

1. wymagania zorganizowane w TO DO listę focusują zespół na zadaniach do wykonania i odhaczenia, a US utrzymują zespół w skupieniu na rozwiązywaniu problemów dla prawdziwych użytkowników.
2. Po zdefiniowaniu celu zespół ma za zadanie wspólne wypracowanie najlepszego sposobu obsługi użytkownika i osiągnięcia tego celu
3. Historie zachęcają zespół do krytycznego i twórczego myślenia o tym, jak najlepiej rozwiązać cel końcowy.
4. Każda mijająca historia, powinna nieść ze sobą poczucie małej wygranej i napędzać zespół - czy tak się faktycznie dzieje zależy mocno od tego czy zespół jako całość faktycznie czuje się odpowiedzialny za całość dostarczonej historyjki. W praktyce bywa z tym różnie…

SZABLON USER STORY: AS A USER… I WANT… SO THAT…

Jako [persona]

- dla kogo budujemy dane rozwiązanie?
- czy jest określona grupa docelowe na podstawie której możemy zbudować personę?
- dobrze mieć wyobrażenie tej osoby, poczucie że jest kimś rzeczywistym

Jako sprzedawca, który dodaje nową fakturę, chcę mieć możliwość wprowadzenia najpierw numeru NIP klienta, aby wszystkie pola danych firmy zostały automatycznie uzupełnione, jeżeli dane klienta znajdują się w bazie.

Kryteria akceptacyjne

w polu do wprowadzenia numeru NIP użytkownik może wprowadzić ciąg 10 cyfry z zakresu 0-9

jeśli użytkownik wprowadzi mniej niż 10 dozwolonych znaków zostanie wyświetlony stosowny komunikat

w polu nie można wprowadzić znaków spoza zdefiniowanego zakresu

pole będzie walidowane „w locie”

SZABLON USER STORY: AS A USER… I WANT… SO THAT…

chcę…

- opis zamiaru, intencji persony, tego co chce osiągnąć
- nie funkcje, nie ficzery

Jako sprzedawca, który dodaje nową fakturę, chcę mieć możliwość wprowadzenia najpierw numeru NIP klienta, aby wszystkie pola danych firmy zostały automatycznie uzupełnione, jeżeli dane klienta znajdują się w bazie.

Kryteria akceptacyjne

w polu do wprowadzenia numeru NIP użytkownik może wprowadzić ciąg 10 cyfry z zakresu 0-9

jeśli użytkownik wprowadzi mniej niż 10 dozwolonych znaków zostanie wyświetlony stosowny komunikat

w polu nie można wprowadzić znaków spoza zdefiniowanego zakresu

pole będzie walidowane „w locie”

SZABLON USER STORY: AS A USER… I WANT… SO THAT…

tak aby...

- Jaka jest ostatecznej korzyść którą stara się osiągnąć?
- Jaki jest problem, który chce rozwiązać?

Jako sprzedawca, który dodaje nową fakturę, chcę mieć możliwość wprowadzenia najpierw numeru NIP klienta, aby wszystkie pola danych firmy zostały automatycznie uzupełnione, jeżeli dane klienta znajdują się w bazie.

Kryteria akceptacyjne

w polu do wprowadzenia numeru NIP użytkownik może wprowadzić ciąg 10 cyfry z zakresu 0-9

jeśli użytkownik wprowadzi mniej niż 10 dozwolonych znaków zostanie wyświetlony stosowny komunikat

w polu nie można wprowadzić znaków spoza zdefiniowanego zakresu

pole będzie walidowane „w locie”

SZABLON USER STORY: AS A USER… I WANT… SO THAT…

- **Kryteria akceptacyjne**
  - Definition of done
  - Warunki konieczne do spełnienia, aby US zostało uznane za zakończone

Jako sprzedawca, który dodaje nową fakturę, chcę mieć możliwość wprowadzenia najpierw numeru NIP klienta, aby wszystkie pola danych firmy zostały automatycznie uzupełnione, jeżeli dane klienta znajdują się w bazie.

Kryteria akceptacyjne

- w polu do wprowadzenia numeru NIP użytkownik może wprowadzić ciąg 10 cyfry z zakresu 0-9
- jeśli użytkownik wprowadzi mniej niż 10 dozwolonych znaków zostanie wyświetlony stosowny komunikat
- w polu nie można wprowadzić znaków spoza zdefiniowanego zakresu
- pole będzie walidowane „w locie”

1. DoD jest jasną i zwięzłą listą wymagań, które musi spełniać oprogramowanie, aby zespół mógł je nazwać kompletnym/skończonym. DoD jest integralną częścią wymagań w projekcie agileowym.
2. Kryteria akceptacyjne to nic innego jak DoD, specyficzne dla danej User Story

OMÓWIENIE SZABLONU DOKUMENTU

PRODUCT REQUIREMENTS DOCUMENT (PRD)