



Czym jest User Story?	2
CCC	2
Card	2
Conversation	2
Confirmation	2
INVEST	3
Independent	3
Negotiable	3
Valuable	3
Estimable	4
Small	4
Testable	4
Dobre praktyki	4
Wartość w User Story	4
Załączniki w User Stories	4
Linkowanie User Stories	4
Epiki	5
User Stories a proces	5
User Story Mapa	5
User Stories a wymagania нефункционалне	5
Strona czynna, nie bierna	5
Kto powinien pisać historyjki?	5
Rozbijanie User Stories	5
Kiedy nie stosować User Stories?	6
User Stories a analiza wymagań	6

Czym jest User Story?

User story to wymagania pisane językiem naturalnym, które pokrywają oczekiwania realnych użytkowników.

W trakcie pisania User Story ważne jest więc, aby:

- stosować formę zrozumiałą dla przeciętnego odbiorcy (język naturalny),
- zaczynać od wymagań - oczekiwań, a nie od rozwiązań technicznych,
- dostarczać wartość realnym użytkownikom.

CCC

CCC to popularny skrót od Card, Conversation, Confirmation.

Card

Card to podstawa historyjki. Najczęściej zapisywana w formie:

Jako (ktoś) chciałbym (cel) aby (korzyść).

W offline'owym wariacie pisania user story, taka informacja musiała się zmieścić na jednej karteczce (sticky note).

Odradzam stosowanie klasycznej formy jako tytuł historyjki np. w Jira, bo nagłówki robią się wtedy nieczytelne.

Conversation

Rozmowa na temat User Story.

W efekcie powinna powstać pełniejsza definicja User Story (uzupełnienie Card).

Przykładowe sekcje:

- Tło biznesowe - wprowadzenie do tematu,
- Opis problemu z perspektywy użytkowników,
- Ograniczenia,
- Założenia,
- Propozycje rozwiązań.

W ramach Conversation unikaj opisywania szczegółów implementacyjnych na wczesnych etapach definiowania User Story. Takie informacje powinny pojawić się dopiero przed rozpoczęciem developmentu. Dzięki temu zaoszczędzisz sporo czasu jeżeli ktoś anuluje Story lub system znacząco się zmieni (więc zmieni się i rozwiązanie).

Confirmation

Potwierdzenie, czyli zestaw testów akceptacyjnych.

Testy te najlepiej pisać z perspektywy biznesowej, w oderwaniu od konkretnej implementacji.

Czyli np. taki krok:

- "Wyszukuję dostępne połączenia na trasie", zamiast:
- "Wpisuję miejsce wyjazdu, miejsce docelowe i naciskam przycisk Szukaj".

Dzięki temu niezależnie jak zostanie zaimplementowane User Story, taki test będzie zawsze aktualny.

Testy możesz pisać korzystając z konkretnej notacji jak np. Gherkin lub opisywać scenariusze językiem naturalnym.

INVEST

INVEST to powszechnie przyjęty skrót charakteryzujący dobre User Stories.

Independent

Czyli Story powinny być od siebie możliwe niezależne.

W praktyce nie jest możliwe aby wszystkie historyjki były niezależne, tak aby można było dowolną z nich wyciągnąć i przenieść w inne miejsce w backlogu.

Systemy najczęściej realizują jakieś procesy, do realizacji których sekwencja jest bardzo ważna.

Story nie powinny jednak być zbyt detaliczne. Tzn. w ramach dostarczanej wartości muszą mieć wszystko co jest potrzebne.

Przykładowo, nie rozbijamy zamawiania przejazdu w Uberze na historyjki: wybranie miejsca początkowego, wybranie adresu docelowego, wybranie typu transportu, potwierdzenie.

To wszystko powinna być jedna historyjka, która dostarcza cały mikro-proces. Dzięki temu zrealizowanie historyjki daje konkretną wartość - możliwość wysłania zapytania o przejazd z punktu A do B.

A to jak taki przejazd potwierdzić i obsłużyć, to już temat na inną historyjkę. W ostateczności ktoś może to robić nawet ręcznie.

Negotiable

User Story to rozmowa. Jeżeli wymagania są w 100% określone, to nie ma potrzeby stosowania User Stories.

Dlatego też na początku określone są cele i założenia. A dopiero z czasem doprecyzowywane są wszystkie detale.

W pewnym momencie musi jednak być punkt, w którym dyskusje się zatrzymują, a zaczyna się implementacja.

Valuable

User Story musi przynosić wartość konkretnym użytkownikom. To nie może być po prostu funkcjonalność, którą ktoś wymyślił od tak sobie.

Wartość powinna być precyzyjnie określona i możliwa do zmierzenia. Dzięki temu będzie dużo łatwiej znaleźć rozwiązania, które doprowadzą do celu.

Estimable

Historyjki muszą być na tyle precyzyjnie opisane aby dało się oszacować ich czas realizacji. Estymacje te mogą być relatywne, czyli poprzez porównanie z innymi User Story. Mogą też być podane w godzinach czy dniach.

Jeżeli zespół ma jakieś wątpliwości, to wszelkie pytania, a zwłaszcza odpowiedzi, powinny znaleźć się w treści story (a nie tylko w głowach uczestników).

Small

User Story musi być możliwe do zrealizowania w jednym sprincie. Jeżeli jest za duże, to trzeba je podzielić. Im mniejsze Story tym lepiej, bo kod jest regularnie merge'owany, a zadania są częściej testowane.

Testable

W ramach User Story muszą być zdefiniowane konkretne scenariusze testowe, które potwierdzą, że historyjka została dostarczona.

Jeżeli nie da się określić takich testów, to należy doprecyzować User Story.

Dobre praktyki

Wartość w User Story

Wartość w User Story jest wtedy, kiedy konkretny użytkownik otrzymuje wymierną korzyść. Może to być np. skrócenie czasu wykonywania operacji, ograniczenie kosztów itp.

Samo zrealizowanie funkcjonalności może nie przynieść wartości. A wręcz przeciwnie, może zaszkodzić, jeżeli funkcjonalność nie będzie działała prawidłowo.

Załączniki w User Stories

Załączaj do User Stories wszystko, co pomoże zrozumieć wymagania i potem je zaimplementować. Mogą to być:

- Notatki,
- Makiety,
- Przykładowe zapytania/odpowiedzi,
- Przykładowe dane, itp.

Nie masz pewności, kiedy dane User Story wejdzie do sprintu. Przygotuj historyjki tak, aby móc wrócić do nich za kilka miesięcy.

Linkowanie User Stories

Łącz zależne User Stories, tak aby mieć pełną historię zmian i zależności. Zwykle lepiej jest połączyć nawet luźno zależne Story, niż nie łączyć nic.

Epiki

Opisuj epiki podobnie jak User Story. Skup się na wartości jaką dostarczają. Używaj INVEST do weryfikacji poprawności epików.

User Stories a proces

Dobłą praktyką przed pisaniem User Stories jest przygotowanie procesu. Może to być diagram dowolnego typu, niekoniecznie zgodny z konkretnym standardem. Ważne aby był zrozumiały.

Następnie do każdego kroku w procesie utwórz epik lub historyjkę.

W ten sposób nie zapomnisz o żadnym ważnym elemencie.

User Story Mapa

User Story Mapa działa podobnie jak proces. Tylko zawiera więcej detali, dzięki czemu łatwiej przejść z USM na pisanie historyjek.

Zapraszam Cię na storymapping.pl gdzie znajdziesz link do bezpłatnych materiałów na temat tej techniki.

User Stories a wymagania niefunkcjonalne

Wymagania niefunkcjonalne też mogą być zapisane jako User Stories. Użytkownikom zależy aby system działał szybko, był intuicyjny czy bezpieczny.

Do niefunkcjonalnych User Stories stosuj te same zasady co do funkcjonalnych.

Strona czynna, nie bierna

W User Stories najważniejszy jest użytkownik. Jeżeli opisujesz wymagania korzystając ze strony biernej (coś jest, staje się itd.) to wtedy pomijasz rolę użytkownika. Dobrym sposobem na opisywanie User Stories jest pilnowanie się, aby stosować wyłącznie stronę czynną.

Kto powinien pisać historyjki?

Pisanie wymagań w formie User Stories wymaga wcześniejszego przygotowania. Bez tego łatwo popaść w słowotok i wyprodukować dużo tekstu o niskiej wartości merytorycznej.

Jeżeli znasz kogoś, komu przydałby się ten materiał do podniesienia kompetencji w zakresie pisania User Stories, to podeślij dalej [link do kursu](#).

Dziękuję :-)

Rozbijanie User Stories

Na wczesnym etapie trzymaj User Stories na wysokim poziomie ogólności. Nie rozbijaj ich na drobne zadania. Podobnie z epikami.

Jeżeli masz wiele drobnych historyjek w backlogu to bardzo ciężko tym zarządzać. Dużo prościej jest przenieść, zmienić czy anulować jedno większe niż 10 mniejszych elementów.

Rozbijaj User Stories na mniejsze tylko wtedy, kiedy jest to niezbędne. Czyli zwykle w trakcie przygotowywania zadań do sprintu.

Myśl o User Stories jak o kamieniach. Jeżeli są duże, to łatwo nimi zarządzać, przenieść z miejsca na miejsce. Dużo trudniej jest zrobić coś pożytecznego z takiego dużego kamienia. Dlatego trzeba go rozbić na mniejsze. Tylko, że takie mniejsze kamienie łatwo zgubić lub wymieszać z innymi. Dlatego rozbijaj tylko tyle ile potrzebujesz.

Kiedy nie stosować User Stories?

Nie stosuj User Stories jeżeli zakres jest precyzyjnie określony i wynika np. ze zmian prawnych czy wymagań integracyjnych. W takim przypadku pisanie User Stories będzie niepotrzebną stratą czasu.

Zwyczajnie opisz co jest do zrobienia i przekaz do implementacji.

Bez dyskusji z całym zespołem na temat pomysłów, rozwiązań itp. Czyli bez całego Conversation.

User Stories a analiza wymagań

Chociaż w User Stories opisujesz wymagania językiem naturalnym z perspektywy użytkownika, to wcześniej czy później konieczne jest przeprowadzenie solidnej analizy.

Do sprintu nie powinny trafiać wymagania z błędami i brakami tylko dlatego, że umówiliśmy się na stosowanie historyjek użytkownika.

Kto i kiedy taką analizę wykonuje to już kwestia do indywidualnego ustalenia.