



Podcast „Menedżer Plus“ – Odcinek 28

Opublikowany: 01 lutego 2016 r.

8 powodów dla których warto pisać historyjki użytkownika z zespołem

Kiedy myślicie o wdrożeniu Scruma w swoim projekcie musicie liczyć się z tym, że czeka Was całkiem sporo różnego rodzaju, ja to nazywam, wywrotek mentalnych. Trzeba się oduczyć pewnych przyzwyczajzeń, pewnych nawyków, które mamy, a które wynikają z naszego dotychczasowego podejścia do pracy w projektach, do pracy związanych z rozwojem produktu.

Jedną z takich wywrotek mentalnych, którą dzisiaj chciałbym poruszyć to jest praca nad wymaganiami, praca nad zbieraniem i dokumentowaniem wymagań. Im szybciej uda Wam się tutaj oduczyć pewnych zachowań tym lepiej.

Przede wszystkim ta wywrotka mentalna, jeśli chodzi o wymagania i zbieranie wymagań dotyczy osoby Product Ownera, który jest głównym dostarczycielem wymagań w projektach scrumowych, jest częścią zespołu scrumowego. To jest duża zmiana w porównaniu do tego, co było wcześniej, do tradycyjnych podejść.

W tradycyjnych projektach klient nie był częścią zespołu wytwórczego, nie był częścią zespołu, który rozwijał produkt. Określał swoje oczekiwania na początku, potem powstawał najczęściej opasły kilkusetstronicowy word i później zespół był już zdany najczęściej tylko na siebie.

W Scrumie wygląda to zupełnie inaczej. Product Owner jest pełnoprawnym członkiem zespołu scrumowego i to jest bardzo duża, poważna zmiana właśnie mentalna, która oczywiście nie jest łatwa i dla Product Ownera, i dla członków zespołu deweloperskiego. Mija najczęściej kilka sprintów zanim ludzie zdążą się z tym oswoić i zaczną tak naprawdę normalnie funkcjonować.

Jeśli chodzi o zwinne zarządzanie wymaganiami to tutaj nacisk w projektach agile'owych położony jest przede wszystkim na konwersację, na rozmowę o wymaganiach oraz o pomysłach związanych z rozwojem danego produktu.

Chodzi o to, żeby bardziej o wymaganiach mówić, rozmawiać niż je pisać. To jest kolejna zmiana mentalna w podejściu do zbierania i dokumentowania wymagań. Stąd w projektach zwinnych pojawia się koncepcja historyjki użytkownika – używana też często za pomocą angielskich słów user stories – jako taki łatwy, szybki sposób zapisywania wymagań klienta.

Te historyjki użytkownika początkowo miały postać takich niewielkich papierowych karteczek, kartoników – różnie to wyglądało – które przypominały trochę fiszki, jak dawniej mogliście na pewno spotkać w katalogu bibliotecznym. Nie wiem, jak to jest teraz, bo dawno nie byłem w bibliotece. Fiszki, jak część z Was zapewne pamięta chociażby ze studiów, zawierają określone hasła, krótkie informacje, notatki.

Dokładnie taka idea przyświeca historyjkom użytkownika. To mają być takie małe karteczki, kartoniki na których umieszczony jest zwięzły opis funkcjonalności, która jest szczególnie pożądana dla określonego klienta, dla użytkownika systemu.

Ważne w tym wszystkim jest to, żeby nie traktować – a niektórzy tak robią – historyjek użytkownika jak jakiegoś takiego zamiennika dla specyfikacji. Historyjka jest przede wszystkim narzędziem komunikacji, współpracy całego zespołu scrumowego, a więc właściciela produktu, Scrum Mastera i zespołu deweloperskiego. A co za tym idzie – Product Owner i zespół powinni wspólnie rozmawiać o historyjkach użytkownika i także wspólnie je pisać.

Pomysł na dzisiejszą audycję wziął się stąd, że bardzo niewielu Product Ownerów faktycznie angażuje zespół, czyli analityków, programistów czy testerów w pisanie tych historyjek – taka jest przynajmniej moja obserwacja w tych projektach z którymi miałem okazję i ciągle pracuję.

Product Ownerzy raczej wolą to robić sami w pojedynkę i też zespół oczekuje trochę od Product Ownerów tego, że te wymagania chociażby w formie historyjek – to nie ma znaczenia tak naprawdę – będą im przekazane na początku sprintu i później „odebrane” (choć bardzo nie lubię tego określenia) na koniec podczas przeglądu.

Dzisiaj podam Wam 8 bardzo konkretnych powodów dla których warto to jednak robić wspólnie, dla których warto pisać historyjki użytkownika wspólnie z zespołem deweloperskim, a więc z analitykami, z programistami i testerami podczas tak zwanych sesji pielęgnacyjnych czy określanych także jako sesje refinementowe, a wcześniej sesje groomingowe.

Pierwszy powód – takie podejście sprzyja bardzo mocno budowaniu zespołu. Każda wspólna praca integruje. Product Owner, który angażuje wszystkich członków zespołu w pisanie historyjek użytkownika pokazuje, że traktuje ten zespół, ludzi, którzy pracują w tym zespole jak takich równoprawnych partnerów.

Nie ma już podziału my-oni, tak jak było to praktykowane w tradycyjnych projektach, jesteśmy. W Scrumie Product Owner, Scrum Master i zespół deweloperski są jednym bytem, jednym organizmem – są właśnie zespołem.

Ale muszę Wam powiedzieć, że to traktowanie Product Ownera jako takiego równoprawnego czy pełnowartościowego członka zespołu na początku wcale tak łatwo ludziom nie

przychodzi. To jest też normalne, oduczamy się pewnych wcześniejszych przyzwyczajzeń, gdzie jednak ten podział był dość mocno zaznaczony.

Kiedyś w jednej firmie pamiętam, że podczas warsztatów robiliśmy taką symulację sesji pielęgnacyjnej w sprincie, czyli takiej sesji refinementowej. Byli na tej sesji zgromadzeni zarówno ludzie z biznesu, a więc przyszli Product Ownerzy, byli też analitycy, programiści, testerzy, a więc członkowie, przyszli członkowie zespołu deweloperskiego.

Ta symulacja trwała około 30 minut i przez połowę czasu rozmowa między biznesem i IT wyglądała mniej więcej tak, że wszystko kręciło się wokół jednego zdania – to wy powiedzcie co konkretnie chcecie, a my wam to zrobimy.

W zespole i przedstawicielach biznesu działały podczas tej symulacji dokładnie takie same mechanizmy, które spotykamy czy spotykaliśmy właśnie w tradycyjnych projektach. To myślenie takie podziałowe mocno – my-oni. My jesteśmy odpowiedzialni tylko za dostarczanie wymagań, a oni są odpowiedzialni za ich realizację.

Muszę Wam powiedzieć, że w tej symulacji bardzo długo żadna ze stron – praktycznie dopiero pod koniec się to zmieniło – ale tak to przez większą część tego ćwiczenia żadna ze stron nie potrafiła tak naprawdę ze sobą współpracować, nie czuli się zespołem. Dopiero w ostatnich minutach symulacji coś w nich pękło i zobaczyli, że tak się nie da. Że to nie chodzi o to, żeby wytwarzać ten sztuczny czy nienaturalny podział, bo potrzebna jest współpraca, żeby odnieść sukces.

Powód numer 2 – wzrasta poczucie odpowiedzialności za produkt. Bardzo ważny argument za tym, żeby wspólnie pisać historyjki użytkownika. Jeżeli zaangażujecie ludzi w jakąś pracę zobaczycie, że bardzo szybko zaczną czuć się współwłaścicielami tego, co robią, tego, co powstaje, tego, co wytwarzają. Scrum jest taką metodą pracy dzięki której zespoły wytwórcze stają się bardziej odpowiedzialne za produkt.

W tradycyjnych podejściach ta odpowiedzialność jest można powiedzieć bardzo rozmyta, bo zespoły pracują w takich silosach, są mocno od siebie oddzielone i tracą tę szerszą perspektywę – perspektywę końcowego produktu właśnie, bo na dobrą sprawę jak sobie popatrzycie, jak ta praca w projektach takich sekwencyjnych i kaskadowych wygląda to dla każdego zespołu produktem jest trochę co innego.

Dla analityków to jest dobre przygotowanie analizy i dobra specyfikacja, dla deweloperów, dla programistów produktem jest dobrze napisany kod, dla testerów dobrze przeprowadzone testy, a więc testy, które wykryją jak najwięcej błędów w tym kodzie itd.

Natomiast ta perspektywa całego produktu w podejściu kaskadowym dość mocno się rozmywa i ludzie tego pełnego obrazku nie widzą. Ta perspektywa produktu końcowego gdzieś znika. W Scrumie z kolei ludzie bardzo mocno czują się współwłaścicielami tego, co wytwarzają.

To jest zupełnie inaczej niż w projektach kaskadowych tradycyjnych. Jeżeli Product Owner czy Product Ownerzy zaangażują faktycznie członków zespołu deweloperskiego w tworzenie i rozwijanie backlogu to automatycznie poczucie odpowiedzialności zespołu wzrasta do kwadratu. Analitycy, programiści i testerzy wiedzą, że ich głos się liczy, że mają wpływ na to w jakim kierunku będzie rozwijał się produkt – to już jest ich produkt.

Powód numer 3 – wzrasta zaangażowanie zespołu. Ludzie oprócz tego, że czują się właścicielami tego, co budują czy rozwijają robią to z prawdziwą pasją i zaangażowaniem, czują, że to jest ich produkt, czują, że mają na ten produkt realny wpływ, że ktoś pyta o ich zdanie, że z nimi rozmawia, traktuje ich jak takiego pełnowartościowego partnera.

Nawet jeżeli członkowie zespołu nie mają odpowiedniej biznesowej wiedzy o produkcie, o klientach czy oczekiwaniach użytkowników – sama jednak możliwość tej dyskusji, bycia jej częścią, uczestniczenia w rozmowie z Product Ownerem robi naprawdę dużą robotę.

Powód numer 4 – wzrasta wiedza o produkcie. Dzięki tym, że Product Ownerzy angażują zespół w pisanie historyjek, programiści, analitycy, testerzy mogą lepiej poznać produkt, mogą lepiej poznać oczekiwania użytkowników, klienta.

I im bardziej będziecie angażować zespoły w tworzenie back logu produktu, tym większa będzie wiedza na jego temat. Tutaj mówię bezpośrednio do Product Ownerów, którzy słuchają tego odcinka.

Znam wiele zespołów scrumowych, w których sporo najlepszych pomysłów na to, jak wygląda dzisiaj produkt pochodziło właśnie od zespołu. Dlaczego z tego nie skorzystać?

Powód numer 5 – wsparcie dla Product Ownera. Product Ownerzy w większości wypadków nie są osobami technicznymi i nie widzą pełnego obrazka, jeśli chodzi o wytwarzanie produktu. Ale to jest normalne – nikt nie oczekuje od nich, że będą omnibusami, będą się znali na wszystkim i że będą potrafili biznesowo czuć produkt, klienta, użytkowników i z drugiej strony będą potrafili na przykład programować.

Na czym polega to wsparcie Product Ownera? Product Ownerzy mogą być na przykład bardzo dobrze zorientowani w różnego rodzaju zależnościach biznesowych czy funkcjonalnych między historyjkami użytkownika.

Jednak już pewnych zależności technologicznych mogą być zupełnie nieświadomi. Tutaj właśnie wsparcie zespołu może być nieocenione. Wiele razy spotykałem w firmach takie sytuacje, że Product Owner układał sobie jakąś określoną kolejność, listę historyjek w backlogu.

A później, gdy rozmawiał z zespołem nagle okazywało się, że musi wszystko pozmieniać, że cała ta kolejność trochę ma się nijak do tego co kryje się za tymi historyjkami, jeśli chodzi

właśnie o zależności technologiczne, o technologicie, które są ze sobą bardzo mocno powiązane.

Zespół przydaje się także Product Ownerowi, jeśli chodzi o pisanie historyjek niefunkcjonalnych – takich, które dotyczą na przykład wydajności, bezpieczeństwa, użyteczności.

Product Owner wie, że nie jest sam, że ma świetnych ludzi na których może liczyć i którzy zawsze mu przyjdą z super pomocą, może zawsze coś skonsultować, obgadać swoje pomysły, które chodzą mu po głowie. No i uzyskać taką szczerą informację na temat tego, co robi, w jakim kierunku chciałby rozwijać swój produkt.

Powód numer 6 – wsparcie przy definiowaniu kryteriów akceptacyjnych. Na dobrą sprawę można by ten powód podciągnąć pod to, o czym mówiłem wcześniej, czyli pod wsparcie Product Ownera.

Ale celowo wyodrębniłem te kryteria akceptacyjne i pomoc przy definiowaniu tych kryteriów akceptacyjnych jako odrębny punkt. Historyjki użytkownika zostały wymyślane po to, żeby – tak jak powiedziałem wcześniej – bardziej rozmawiać o wymaganiach, bardziej o nich gadać niż je pisać.

Jednym z częstych mitów na temat metod zwinnych jest to, że w tym całym agile’u nie ma żadnej dokumentacji, że dokumentacja jest niepotrzebna. Dokumentacja jest, ale dokumentujemy tylko to, co jest nam faktycznie potrzebne i w możliwie najbardziej odchudzonej formie, bo tylko wtedy te dokumenty będą nam dostarczały faktycznie jako taką wartość.

Tutaj z pomocą przychodzą nam właśnie historyjki użytkownika. Jest to genialne narzędzie do tego, żeby zapisać tylko najbardziej istotną informację o tym co trzeba zrobić, żeby zapisać taką esencję, która wiąże się z wdrożeniem danej funkcjonalności. Wszystkie szczegóły natomiast są zapisywane w formie tzw. kryteriów akceptacyjnych.

Jeszcze zanim firmy nasze zostały opanowane przez narzędzia elektroniczne, pamiętam jak na początku bardzo często stosowaliśmy w projektach i w zespołach user story – historyjki użytkownika w formie po prostu karteczek, które chowało się do pudełka po butach.

Po zakończonym sprincie to w trakcie rozmowy z Product Ownerem na odwrocie tej karteczki było miejsce na to, żeby zanotować szczegóły tej rozmowy. A te szczegóły to właśnie nic innego jak kryteria akceptacyjne, takie historyjki użytkownika. Dzięki nim zespół mógł lepiej zrozumieć co tak naprawdę należy zbudować i w jaki sposób powinno to być przetestowane.

Kryteria akceptacyjne to jest też taki dobry scenariusz dla zespołu do tego, żeby przeprowadzić, no i przygotować się wcześniej do tego, żeby zrobić przegląd sprintu. Muszę Wam powiedzieć, że Product Ownerzy, tak jak obserwuję w różnych firmach, nie bardzo kwapią się do tego, żeby te kryteria akceptacyjne faktycznie dla historyjek określać, definiować. No bo to są już szczegóły.

Wielu Product Ownerów myśli, że jak hasłowo napisze na historyjce co ma być zrobione to tak na dobrą sprawę już wystarczy. A tak nie jest. W kryteriach akceptacyjnych jest to całe, że tak kolokwialnie powiem, mięso. Tam są konkretne funkcjonalne oczekiwania na podstawie których możemy powiedzieć, czy dane wymaganie zostało faktycznie zaimplementowane zgodnie z oczekiwaniami Product Ownera czy nie.

Tutaj zespół jest nieocenionym wsparciem dla Product Ownera przy definiowaniu tych kryteriów – pomaga je nazwać, pomaga je dookreślić. Ale też – zwłaszcza w początkowych sprintach – jak zespół startuje ze scrumem wiele razy widziałem, że zespół dba o to, jest taką drugą parą oczu można powiedzieć dla Product Ownera, żeby te kryteria faktycznie się definiowały, żeby faktycznie powstawały.

Powód numer 7 – wzrasta kreatywność. Mówi się, że co dwie głowy to nie jedna. Takie wspólne pisanie historyjek niesamowicie wyzwala kreatywność w zespole. Analitycy, programiści, testerzy to kopalnia wiedzy i pomysłów związanych z rozwijaniem produktu. Product Owner – już mówiłem to kilka razy dzisiaj – nie jest sam.

Kiedy pojawi się jakiś problem może zawsze liczyć, że pojawi się też kilka możliwych rozwiązań. Zespół deweloperski to też różnorodność – różnorodność pomysłów, różnorodność kompetencji, różnorodność ról, bo tak zespół właśnie działa. To też bardzo dobrze służy, jest bardzo dobrym katalizatorem kreatywności i to się także przekłada bardzo fajnie na rozwój samego produktu.

Powód numer 8 – to już jest ostatni powód, który dla Was przygotowałem – to wcale nie kosztuje tak dużo, jak się Wam wydaje. To jest taki argument, z którym się dosyć często spotykam kiedy zachęcam Product Ownerów do tego, żeby angażowali zespół właśnie w pisanie historyjek użytkownika.

Często słyszę takie argumenty, że tyle jest spotkań w naszej firmie, nie mamy czasu na kolejne, godzinne spotkanie, przecież z udziałem 8 osób to już nie jest 1 godzina tylko 8 godzin itd. Te argumenty słyszę zarówno od strony Product Ownerów, ale także ze strony zespołu deweloperskiego.

Oczywiście, to jest prawda. Oczywiście, muszę się zgodzić z tym, że takie spotkania – zresztą jak każde inne spotkania w firmie – kosztują. Ale z tym argumentem jest trochę tak jak z delegowaniem zadań.

Wielu liderów nie chce delegować zadań, bo twierdzi, że wdrożenie kogoś w temat zajmie mu więcej niż realizacja tego zadania, w związku z tym woli to zadanie zrobić sam, a nie tracić czasu na to, żeby daną osobę wprowadzić w temat, nauczyć obsługi narzędzia, powiedzieć o co chodzi itd.

I faktycznie tak jest. Jak sobie popatrzycie w mikroskali na delegowanie zadań to tak wygląda. Delegowanie zadań faktycznie może trochę więcej kosztować na początku, ale później nam się to zwraca.

Podobnie jest tutaj – jeżeli zaangażujecie zespół właśnie w pisanie historyjek to w dalszej perspektywie czasowej na pewno Wam się to opłaci. Zobaczycie, że ludzie będą zadawali Wam mniej pytań na temat produktu, będą mniej pytać o oczekiwania użytkowników, klientów.

Bo siłą rzeczy, jeżeli będziecie wspólnie z nimi nad produktem, nad rozwojem produktu pracować to też będą te oczekiwania dużo lepiej rozumieć. Nie będą za każdym razem potrzebowali angażu Product Ownera, żeby znaleźć odpowiedź na pytanie czy na rozwiązanie danego problemu. Docelowo na pewno na tym więcej zyskacie.

Przedstawiłem Wam 8 powodów dla których Product Owner i zespół deweloperski powinni wspólnie pracować nad pisaniem historyjek użytkownika. Szybko podsumowując.

Pierwszy powód to było sprzyjanie budowaniu zespołu. Drugi powód to wzrost odpowiedzialności za produkt. Trzeci – wzrost zaangażowania zespołu. Czwarty – wzrost wiedzy o produkcie. Piąty – wsparcie dla Product Ownera. Szósty – wsparcie przy definiowaniu kryteriów akceptacyjnych. Siódmy – wzrost kreatywności. Ósmy – to wcale tak dużo nie kosztuje.

Jeśli o czymś zapomniałem albo uważacie, że powinno się tam znaleźć więcej powodów, które pominąłem to zachęcam Was bardzo mocno do dyskusji pod wpisem do dzisiejszej audycji. Żeby to zrobić wystarczy wejść na stronę mojego bloga mariuszchrapko.com/028 – tam znajdziecie też wszystkie notatki dotyczące tego odcinka.

Ostatnio zauważyłem, że jest też Was coraz więcej na moim facebookowym fanpage’u, co mnie niezmiernie cieszy. Jestem tam dla Was codziennie dostępny i od jakiegoś czasu systematycznie wrzucam duże dawki darmowej wiedzy, porad i różnego rodzaju inspiracji.

Dlatego szczególne wielkie dzięki dla tych spośród Was, którzy znajdują czas na lajkowanie i udostępnianie publikowanych tam treści i od czasu do czasu znajdują czas, żeby pogadać ze mną w komentarzach.

A jeżeli podobał Wam się dzisiejszy odcinek, zostawcie swoją ocenę w iTunes w formie gwiazdek lub krótkiej recenzji. Przypominam Wam, że te oceny pomagają mi w utrzymaniu

dobrej widoczności podcastu w rankingach dzięki czemu osoby, które jeszcze go nie słyszały mogą do niego łatwiej dotrzeć.

To już wszystko na dzisiaj. Na koniec mam dla Was utwór o sofie i coca-coli – tak przynajmniej zrozumiałem z tej piosenki. Trzymajcie się i do usłyszenia!