

Brakujący element Agile

Część 2: Wprowadzanie feedbacku w życie

W tym artykule przedstawię wyzwania, które napotykamy przy wprowadzaniu feedbacku w życie. Opiszę skuteczne metody jego implementacji i sposoby radzenia sobie z często pojawiającymi się problemami. Pokażę przede wszystkim propozycję frameworka, który pomoże Ci podejść do feedbacku „po inżyniersku”, czyli profesjonalnie i ze strukturą.

W poprzednim odcinku serii opowiedziałem, jak efektywnie dawać feedback innym. Feedback stanowi elementarną część metodyki Agile: umożliwia lepszą komunikację z klientem, zwiększanie efektywności zespołu, ulepszanie procesu, iteracyjne tworzenie produktu spełniającego w pełni potrzeby użytkownika. Obecność kultury dawania/otrzymywania feedbacku w zespole to ważny pierwszy krok na drodze do lepszej współpracy. Po wykonaniu tego kroku stajemy przed wyzwaniem – *jak otrzymany feedback wprowadzać w życie?* Przykładowo: co ma zrobić Szymon (młodszy programista), jeśli Kasia (doświadczony tester) powie mu, że oddaje funkcjonalności do testów zbyt szybko – podając jako przykład 8 z 10 historyjek, które wróciły do Szymona ze zgłoszeniem ewidentnych bugów? Szymon może spróbować testować historyjki trochę dokładniej, jednak brak doświadczenia będzie utrudniał mu poprawę. Korzyść, jaka z tego wyniknie dla Kasi, to spokój ducha – Kasia może sobie powiedzieć w myślach: *przynajmniej próbowałam*. Wyzwaniem dla Kasi jest odpowiedzieć na pytanie – jak może pomóc Szymonowi efektywniej testować funkcjonalności przed odaniem jej do QA. Dla Szymona będzie to przecież nie lada problem – jest jeszcze niedoświadczony i trzeba czasu, żeby osiągnął większą skuteczność.

JAKOŚĆ FEEDBACKU

Feedback, który otrzymasz od współpracowników, klientów czy nawet partnera życiowego, będzie jakościowo bardzo zróżnicowany. Czasem dostaniesz konkretną informację, co robisz nieefektywnie i jak to poprawić: *podczas retrospektyw często komentujesz na temat błędów Jacka. Czy mógłbyś zamiast tego porozmawiać z nim osobiście?*. Często jednak feedback będzie dużo bardziej „mętny”: *czuję, że podczas niektórych spotkań jesteś jakby nieobecny*.

Na własnym przykładzie wiem, że kiedy otrzymuję feedback o kiepskiej jakości, moją pierwszą reakcją jest frustracja. Jeśli próba dopytania drugiej osoby o szczegóły nie pomaga, frustracja staje się jeszcze większa. Naturalnym odruchem wydaje się wtedy zignorowanie takiego feedbacku. Wiem jednak na bazie doświadczeń, że każdy feedback może być pouczający i rozwijający. Dlatego staram się nawet taki feedback wykorzystać i znaleźć metody, które pomogą mi go implementować. Jedną z technik, którą stosuję w takich trudnych sytuacjach, jest „skupianie uwagi”.

SKUPIANIE UWAGI

Zdarza się, że nie zgadzasz się z feedbackiem, który otrzymujesz, np. *mało udzielasz się w dyskusjach*. Czasem dotyczy on kwestii, nad którą nigdy się nie zastanawiałeś „kiedy rozmawiasz z innymi, nie odnosisz się do ich emocji”. Wbrew pozorom, to jeden z lepszych rodzajów feedbacku, jaki możesz od kogoś dostać. Dotyczy on cech bądź zachowań, które znajdują się w Twoim „martwym polu”. Są to aspekty, które z Twojej perspektywy uznajesz za nieistotne i tego typu feedback daje szansę na rozwinięcie zupełnie nowej umiejętności. Warto pamiętać, że ciągłe ulepszanie (ang. *continuous improvement*) w metodykach Agile dotyczy procesu, produktu, jak również członków zespołu.

Z takim feedbackiem należy się najpierw oswoić. Do tego celu stosuję technikę „skupiania uwagi”. Polega ona na przyciągnięciu uwagi Twojego umysłu do problemu, starając się zachować podejście bezkrytyczne. Ta metoda opiera się na założeniu, że Twoje myśli lubią podróżować w mózgu popularnymi drogami. Skupienie uwagi pozwala „zachęcić” sygnały do podróży nowymi, nieznanymi ścieżkami. Z podobnej przyczyny potrzebne jest zachowanie bezkrytycznego podejścia. Przyjmowanie istniejących opinii kieruje Twoje myśli na znane szlaki i powstrzymuje od tworzenia się nowych struktur w mózgu. Innymi słowy – powstrzymuje proces uczenia się.

Założmy, że feedback, na którym chcę się skupić, odnosi się do tego, jak rzadko wyrażam swoją opinię podczas spotkań. Skupienie uwagi najłatwiej osiągnąć poprzez zadanie pytania w stylu: *czy ja rzeczywiście za mało się udzielam w dyskusjach*. Podejście bezkrytyczne polega na tym, że unikam oceny, tj. *czy jestem lepszy, bo to robię* albo *czy jestem gorszy, bo tego nie robię*.

Dobrym sposobem jest zapisywanie każdego pytania – najlepiej w miejscu, na które często zwracasz uwagę. Wtedy częściej utrwalasz nowe ścieżki i skupiasz uwagę umysłu na interesującej Cię kwestii. W rozważanym przypadku należy co najmniej przypominać sobie je przed każdym spotkaniem, aby zainicjować w Twoim mózgu proces zbierania informacji oraz uczenia się. Natomiast po każdym spotkaniu warto, byś zastanowił się, czego nowego się nauczyłeś, lub nawet utrwalił swoje myśli w formie pisemnej (o której wspomnę jeszcze w sekcji „journaling”).

SAMOŚWIADOMOŚĆ

Częstym błędem popełnianym przy implementacji feedbacku jest przyjęcie założenia, że albo dokonamy zmiany, albo zostaniemy przy obecnym zachowaniu. Tymczasem istnieje jeszcze trzecia droga, czyli bycie świadomym swojego zachowania. Najlepiej wytłumaczyć to przez przykład. Kierownikiem projektu w jednym z zespołów, gdzie pracowałem, był Marcin. Już pierwszego dnia po dołączeniu do zespołu Marcin poinformował nas, czego możemy się po nim spodziewać. Powiedział, że często wygląda, jakby nadmiernie się przejmował sytuacją w projekcie, i podkreślił, że wynika to z jego osobowości. Taka informacja jest bardzo cenna i pomogła nam zachować spokój, kiedy Marcin wyglądał na mocno przejętego. Zrozumienie i dobra komunikacja jest fundamentem Agile, zwłaszcza między kierownikiem projektu i członkami zespołu programistycznego.

Marcin z pewnością zebrał wiele razy feedback dotyczący tego zachowania. Wiedział również o trzeciej drodze, czyli jak może pomóc mu samoświadomość. Jest to cecha, którą bardzo cenimy u innych. Pozwala uniknąć konfliktów przez błędną interpretację czyjegoś zachowania. Ludzie mający większą samoświadomość cieszą się również większym zaufaniem. Znajomość swoich wad i zalet pozwala im na większą efektywność.

Po otrzymaniu feedbacku w niektórych sytuacjach możesz zastanowić się, czy możesz wybrać „trzecią drogę”. Powinieneś wtedy wykorzystać świadomość swojego zachowania, aby zwiększyć efektywność. Jedną z technik zaprezentowałem już na przykładzie historii Marcina. Polega ona na poinformowaniu innych o swoim wzorcu zachowania, aby byli lepiej przygotowani. Jestem pewien, że pracując z feedbackiem, znajdziesz wiele innych metod, które lepiej umożliwią Ci wykorzystanie samoświadomości.

WPROWADZANIE ZMIANY

Trzecim sposobem, któremu również poświęcę najwięcej miejsca w artykule, jest wprowadzenie zmiany na bazie otrzymanego feedbacku. Bardzo często wprowadzanie zmiany odbywa się w sposób chaotyczny. Równie często, z tego właśnie powodu, utrudnia osiągnięcie celu. To może generować znudzenie, frustrację, a nawet złość. Wielokrotnie zdarzyło mi się widzieć osoby, które próbują przeprowadzać kilkanaście zmian dotyczących swojego zachowania i poddają się przytłoczone ich ogromem. Wprowadzanie zmian w życie zawodowe, zmian dotyczących swojego zachowania, reakcji, wzorców komunikacji jest takim samym projektem jak projekty programistyczne, które prowadzimy na co dzień. Dlatego proponuję framework oparty o uproszczony cykl życia projektu w metodyce Agile.

budowanie backlogu feedbacku

Kiedy zaczynasz zbierać feedback, bardzo szybko okazuje się, że liczba rzeczy, nad którymi mógłbyś pracować, przerasta Twoje możliwości. Czasem feedback od jednej osoby to już za dużo, żeby wdrożyć wszystkie zmiany naraz. Dlatego listę rzeczy, które rozważasz do zmiany, warto przechowywać w jednym miejscu jako „backlog” do implementacji. Oczywiście zaleta to posiadanie explicite listy rzeczy, które mogą poprawić Twoją efektywność. Ponadto jest ważną deklaracją: *wiem, że są aspekty, gdzie jestem mniej efektywny, ale w tym momencie skupiam się na tych istotniejszych*. Na etapie budowania backlogu feedbacku możesz dopisywać feedback o różnej „szczegółowości” „słuchać uważnie na standupie” oraz „poszerzyć wiedzę na temat baz noSQL”. Warto jednak zapisać go w sposób wystarczająco zrozumiały – możesz przecież zdecydować o wdrożeniu go w życie po kilku miesiącach od wpisania na listę.

iteracje, eposy¹ i historyjki

Z doświadczenia wiem, że zarówno od siebie, jak i od innych oczekujemy błyskawicznych zmian. W rzeczywistości zmiany są jednak powolne i wymagają wykonywania małych kroków. Jest bardziej niż prawdopodobne, że większość „historyjek” w Twoim backlogu feedbacku to w rzeczywistości eposy – feedback, wymagający dużych zmian, które mogą trwać wiele miesięcy. Próba zaadresowania zmiany sformułowanej w postaci eposu może być bardzo trudna i powodować frustrację. Dlatego zmiany warto wprowadzać w iteracjach o określonej długości, np. 2 tygodnie. W przeciwieństwie do metodyk Agile w przypadku feedbacku nie potrzeba rozbić eposu na historyjki. Zamiast tego warto się zastanowić, jaki cel jest osiągalny w ciągu kolejnych dwóch tygodni, który jednocześnie przybliży Cię do zrealizowania eposu. W ten sposób za każdym razem projektujesz historyjkę na jedną iterację, a w perspektywie czasu przybliżasz się do realizacji swojego celu².

planowanie iteracji

Dobry plan iteracji feedbacku wymaga efektywnie zaprojektowanych historyjek. Bardzo przydatnym narzędziem jest technika GROW. Akronim GROW rozwija się jako Goals (pol. cele), Reality (pol. stan obecny), Options/Obstacles (pol. opcje/ograniczenia), Way forward (pol. dalsze kroki). W dalszej części przedstawię, jak zaplanować przykładową historyjkę feedbacku w oparciu o tę technikę. Jako przykład historyjki posłuży mi wspomniany wcześniej we wstępie przykład feedbacku. Załóżmy, że w backlogu Szymona znajduje się epos „zgłaszając do testów funkcjonalności pozbawione błędów ocenianych przez testerów jako ewidentne”³.

stan obecny

Celem tego kroku jest dokładne zastanowienie się, jaki jest punkt startowy. Bardzo często zdarza się, że podejmujemy się zmiany bez pełnego zrozumienia, z jakimi problemami zmagamy się teraz, jak daleko od celu się znajdujemy. Wprowadzanie zmiany w ten sposób przypomina „ruchy Browna”⁴ – jest chaotyczne i nie ma określonego celu.

W sytuacji Szymona wartościowa informacja znajduje się w feedbacku od Kasi – 8 na 10 historyjek w ostatniej iteracji zawierało ewidentne błędy. Szymon zna liczbę historyjek, co pozwala mu na lepsze zaplanowanie celu. Kasia wspomniała również o ewidentnych błędach – pomocne byłoby dopytać ją, co przez to rozumie. Szymon powinien również zastanowić się, co według niego może być przyczyną takiego stanu rzeczy, np. zmiany w ostatniej chwili albo mało testów jednostkowych. Załóżmy, że Kasia podała jako przykład ewidentnych błędów wyjątki rzucane przez aplikację. Szymon natomiast ocenił, że zaniedbuje testy jednostkowe negatywnych ścieżek.

cel

Mając zdefiniowany stan obecny, możesz przejść do określenia, co jest celem zmiany. W przygotowaniu efektywnych celów pomoże Ci SMART – jedna z nielicznych koncepcji reprezentowanych przez akronim, które są przydatne. Ich spełnienie pozwala Ci budować motywację potrzebną do długoterminowej zmiany. Jej szczegółowy opis wraz z wytłumaczeniem przeznaczenia znajdziesz poniżej.

» ***Specific (precyzyjny)***

Cel powinien skupiać się na konkretach, np. *opanuję techniki proponowane w książce Kenta Becka o TDD zamiast nauczę się TDD*. Z dobrze zdefiniowanym celem łatwiej wymyślić kolejne kroki prowadzące do niego oraz utrzymać motywację. Trudniej też oszukać samego siebie, mówiąc np. *No tak, uczyć się TDD, wczoraj w kuchni uważnie słuchałem, jak Maciek opowiada o TDD*.

» ***Measurable (mierzalny)***

Powinieneś zdefiniować cel tak, abyś mógł mierzyć postępy w jego realizacji. „Będę mniej arogancki wobec klienta” jest przykładem celu, który jest mierzalny, pod warunkiem że uda Ci się uzyskać po każdym spotkaniu feedback odnośnie Twojego zachowania (niekoniecznie musisz podawać mu do wypełnienia profesjonalny kwestionariusz z pytaniem: *Jak arogancki byłem w skali 1 do 5?*).

» ***Achievable (osiągalny)***

Łatwo się zniechęcić, jeśli przez kilka kolejnych iteracji nie osiągniesz założonego celu.

Cel powinien być zdefiniowany tak, aby można go było osiągnąć w ciągu jednej iteracji. Zamiast *nauczę się GITa* lepiej postawić sobie cel: *zrozumiem, które komendy GITa są odpowiednikami komend SVNa*.

» ***Relevant (mający znaczenie)***

Cel musi mieć znaczenie dla Ciebie i w kontekście Twojej pracy. Bez spełnienia tego warunku trudno o motywację. Załóżmy, że kierownik projektu prosi Cię, abyś rzadziej jadł obiady przy biurku. Jeśli się z nim nie zgadzasz, powinieneś najpierw zrozumieć jego potrzeby i zdefiniować taki cel, który zadowoli potrzeby obu z Was.

» ***Time-related (ograniczony czasowo)***

Trudno będzie Ci utrzymać motywację, jeśli Twoim celem będzie *nauczę się Emacsa* bez określania ram czasowych. Stwierdzenie *nauczę się Emacsa w ciągu 3 miesięcy* pomaga Ci podkreślić, że ten cel jest dla Ciebie istotny. Zamknięcie go w ramach czasowych zwiększa motywację i poczucie pilności w jego realizacji.

Wróćmy więc do Szymona. Na podstawie wcześniej zdefiniowanego stanu obecnego może postawić sobie jako cel „zapewnić, że funkcjonalności oddane do testowania nie rzucają nieoczekiwanych wyjątków”.

1 ang. *epics*

2 Przedstawiony sposób jest bazowany na koncepcji Solutions Focus opracowanej przez Marka McKergow. Więcej informacji na blogu guru Agile Alistaira Cockburna: <http://alistair.cockburn.us/Solutions+Focus+aka+Delta+Method>

3 Szymon zastosował dobrą praktykę wprowadzania zmian, czyli formułowanie ich w sposób pozytywny, tj. „odnosić się w sposób kulturalny do klienta” zamiast „nie odnosić się arogancko do klienta”

4 ruchy Browna – chaotyczne ruchy cząstek w płynie

ograniczenia

Następny krok polega na zidentyfikowaniu, jakie przeszkody oraz trudności oddzielają Cię od celu. Dopiero kolejnym etapem jest opracowanie opcji, które pozwolą je pokonać. Od zdefiniowania ograniczeń warto zacząć z dwóch powodów. Po pierwsze, jest to krok łatwiejszy – ludziom łatwiej myśleć negatywnie, do unikania zagrożeń przystosowała nas ewolucja (konsekwencje ugryzienia przez węża chowającego się w krzaku jeżyn są dużo większe niż korzyść z zebrania jeżyn). Po drugie, problemy to coś, co już istnieje – wystarczy na ogół przywołać ich przykłady, podczas gdy rozwiązania to coś, co musisz dopiero wymyślić.

Lista ograniczeń z perspektywy Szymona to:

- » kod, który oddaje do testów, rzuca niespodziewane wyjątki
- » zarządzanie wyjątkami w moim kodzie jest podatne na błędy

opcje

Z perspektywy GROW ograniczenia są przeszkodami, które oddzielają Cię od celu, natomiast opcje sposobami na ich pokonanie. Po wypisaniu wszystkich ograniczeń możesz zastanowić się, czy jeśli usuniesz je wszystkie, cel zostanie osiągnięty. GROW zakłada, że po pokonaniu wszystkich ograniczeń powinieneś osiągnąć cel.

Lista opcji w sytuacji Szymona:

- » zapewnić, że wyjątki są obsługiwane
- » ulepszyć sposób zarządzania wyjątkami w kodzie

dalsze kroki

Opcje mogą mieć formę abstrakcyjnych koncepcji, dlatego musisz je „przekonwertować” do konkretnych kroków. Na przykładzie Szymona będą to:

- » przed oddaniem kodu do testów sprawdzić pokrycie kodu z wykorzystaniem Cobertury
- » poprosić jednego ze starszych programistów o ocenę mechanizmu zarządzania wyjątkami i propozycję ulepszeń

WZORCE IMPLEMENTACJI FEEDBACKU

Podobnie jak w przypadku kodu, również w implementacji feedbacku pojawiają się wzorce. Ich znajomość ułatwia wprowadzanie feedbacku w życie i pozwala uniknąć „odkrywania koła na nowo”. Poniżej przedstawiam kilka z tych metod, które uważam za najbardziej przydatne.

journalling

Journalling wymaga od Ciebie prowadzenia notatek na temat wprowadzanej zmiany. Może przybierać wiele form; trzy z nich, które ja uważam za najcenniejsze, to:

- » styl wolny – notujesz dowolne wrażenia z procesu wprowadzanej zmiany w takiej formie, jaka jest dla Ciebie najwygodniejsza.
- » meta-feedback – czyli feedback na temat implementacji feedbacku. Wymaga od Ciebie ewaluacji wprowadzanych zmian, oceniając, co robisz dobrze, a gdzie powinieneś poświęcić więcej czasu i pracy.
- » zapisywanie sukcesów – zbliżony do meta-feedbacku, ale ograniczasz się wyłącznie do zapisywania rzeczy, które idą dobrze. Uważam ten sposób za przydatny, kiedy dopiero zaczynasz zmianę. Dostarcza Ci potrzebnego wsparcia i pomaga wytrwać w początkowych etapach wprowadzania zmiany.

Przydatną techniką przy prowadzeniu notatek jest wprowadzenie ograniczenia czasowego. Ograniczenie czasowe ułatwia Ci zmotywować się do ich częstszego prowadzenia, zgodnie z myślą „to tylko 3 minuty, na tyle przecież mogę znaleźć czas”.

metoda bardzo małych kroków

Bardzo często widzę, jak ludzie wstrzymują się przed implementacją feedbacku, zakładając, że nie są w stanie nic zrobić. Po krótkiej rozmowie okazuje się, że stawiają przed sobą bardzo duże oczekiwania. Warto w takiej sytuacji zastosować „metodę bardzo małych kroków” i zadać sobie pytanie: *Jaka jest najprostsza możliwa rzecz, którą mogę zrobić, żeby znaleźć się bliżej celu.* Załóżmy na przykład, że moim problemem jest gadatliwość podczas spotkań. Najmniejszym możliwym krokiem może być poproszenie jednego z uczestników o zasygnalizowanie, kiedy uzna, że rozmowa przerodziła się w mój monolog. Ten pierwszy krok pozwoli mi w pełni zdać sobie sprawę z tego, jak często mi się taka sytuacja zdarza, i być może z czasem lepiej kontrolować się samemu. Zgodnie z tym, co powiedział G.K. Chesterton: „rzeczy warte zrobienia są warte zrobienia nawet źle”⁵.

quotas

„Quotas” polega na wprowadzeniu określonych minimów dla czynności, które zidentyfikowałeś przy etapie „dalsze kroki”. Załóżmy, że postanowiłeś rzadziej przerywać wypowiedzi innych podczas spotkań. Możesz wtedy określić, że możesz przerwać tylko trzy razy podczas każdego spotkania. Podobnie jeśli uznałeś, że powinieneś częściej się odzywać podczas spotkań, mógłbyś oczekiwać od siebie, że odezwiesz się co najmniej raz. Alternatywą jest wprowadzenie „zliczania”, kiedy notujesz, ile razy odezwałeś się podczas spotkania. Mając te dane, możesz przyjąć oczekiwanie, że ich średnia powinna urosnąć z czasem do zadowalającego Cię poziomu.

PODSUMOWANIE

W artykule opisałem trzy sposoby implementacji feedbacku: skupianie uwagi, samoświadomość, oraz wprowadzanie zmiany. W zależności od konkretnej sytuacji możesz zastosować jeden z nich, albo wszystkie naraz. Warto pamiętać, że zmienianie zachowania jest trudne i pozostałe techniki implementacji feedbacku mogą być równie korzystne. Dla przypadków, kiedy zdecydujesz się na wprowadzenie zmiany, przedstawiłem framework, który pozwoli Ci zrobić to efektywniej. Chcę zaznaczyć, że nie musisz od razu podejmować się wdrożenia go w pełni. Zachęcam do wybrania fragmentów, które w tym momencie uznasz za przydatne. W podobnym celu przedstawiłem wzorce implementacji feedbacku – są to techniki, które sprawdziłem sam, i uznałem jako przydatne.

Wprowadzanie feedbacku w życie jest procesem i z pewnością napotkasz trudności. Informacje, które udzielają Ci o Tobie inni ludzie, są na ogół bezcenne. Sprawne wykorzystanie rad, opinii i uwag od innych z czasem mocno zapoczątkuje.

⁵ ang. “Things that are worth doing are worth doing badly”

Paweł Badeński

pawel.badenski@gmail.com

Do niedawna konsultant w firmie ThoughtWorks, gdzie pracował jako programista, trener oraz coach. Obecnie trener i konsultant w firmie Bottega IT Solutions. Bloguje pod adresem <http://the-missing-link-of-agile.com>. Pasjonat improwizacji teatralnej, psychologii stosowanej i neurobiologii oraz ich zastosowania w kontekście tworzenia oprogramowania.

