

Brakujący element Agile

Część 1: Feedback

W serii „Brakujący element Agile” opowiem o konkretnych i praktycznych technikach miękkich stosowanych przez dojrzałe zespoły Agile. Opisywane techniki są brakującym ogniwem, bez których Agile po prostu nie działa. Dowiesz się o modelu interakcji Satir, modelu NVC Rosenberga, frameworku środowiska myślenia (Thinking Environment®), procesie GROW i innych. Ich zrozumienie pozwala budować zespoły, które rozwiązują napotkane wyzwania, prowadzą zdrową komunikację w ramach zespołu oraz z klientem, w ciągły sposób rozwijają się i ulepszają metody pracy.

WSTĘP DO SERII

Zacznijmy od historii. W dużej firmie informatycznej szefowie postanowili, że trzeba zwiększyć efektywność i wprowadzić Agile. Członków zespołu wysłano na szkolenie z metodyki Scrum. Na 2 miesiące zatrudniono niezależnego Agile coacha. Wprowadzono standupy, estymację, dwutygodniowe iteracje, retrospektywy i wiele innych praktyk. Po 4 miesiącach zauważono widoczne zmiany – klient jest bardziej zadowolony, użytkownicy zgłaszają mniej błędów. Komunikacja z klientem jest natomiast tak samo trudna, w zespole wciąż te same konflikty, a z retrospektyw niewiele wynika. Brzmi znajomo? W wielu zespołach wprowadzenie Agile polega bardziej na rebrandingu zespołu, niż zmianie sposobu pracy. Niewiele mówi się o „brakującym elemencie” – miękkich technikach, które pozwoliłyby realizować założenia manifestu Agile. Jak nauczyć się rozwiązywać konflikty w zespole, zamiast zamiatać je pod dywan? Jak przekonywać szefa do wprowadzenia pair programmingu? Jak zorganizować zespół, którego członkowie nie boją się powiedzieć, że meeting jest bezcelowy? Seria artykułów „Brakujący element Agile” zawiera narzędzia oraz techniki pozwalające odpowiedzieć na te pytania.

WSTĘP

W artykule przedstawię 2 modele komunikacji:

- » model drabiny wnioskowania Argyrisa
- » model interakcji Satir

Na bazie przedstawionych modeli pokażę, jak zrefaktoryzować 3 przykłady kiepskiego feedbacku do postaci produktywniej i przydatnej. Na koniec podam 5 praktycznych rad, które pomogą Ci wprowadzić efektywny feedback do Twojego zespołu.

W jednym z zespołów, gdzie zajmowałem się coachingiem, miałem do czynienia z bardzo dobrym technicznie programistą. Przyjmijmy, że miał na imię John. John był niesamowicie inteligentny i posiadał dużą wiedzę techniczną, dzięki czemu wyróżniał się na tle przeciętnego technicznie zespołu. Podczas sesji pair programmingu John implementował funkcjonalność, a siedząca obok osoba patrzyła się w monitor i czekała na to, jak skończy. Wiele osób w projekcie uważało Johna za osobę nieprzyjemną i dominującą. Miało to negatywny wpływ na jakość komunikacji i nastroje w zespole. Perspektywa takiego stanu rzeczy nie wyglądała ciekawie. Na szczęście kilka osób przedstawiło Johnowi, jak sprawa wygląda. Opisałi mu, jak jego zachowanie jest postrzegane i jak jest odbierane. To pozwoliło mu na poprawę, na czym zyskał zarówno sam John, jak i cały zespół. Wystarczył feedback.

Feedback jest jedną z pięciu wartości, tworzących fundament Extreme Programming. Istnieje wiele praktyk technicznych oraz procesowych, które wykorzystują feedback, m. in. Continuous Integration, Test Driven Development,

Standupy czy Retrospektywy. Zespoły Agile wdrażają te praktyki – samodzielnie, bądź z pomocą trenerów, po czym następuje zastój. Do rozwiązania prawdziwych problemów potrzebna jest efektywna komunikacja, kreatywność, wzajemne wsparcie członków zespołu. A to z kolei wymaga zaufania i zdrowych relacji. Niestety w mainstreamie metodyki Agile te kwestie są często ignorowane na rzecz praktyk i procesów. Mało kiedy dostarcza się konkretnych technik feedbacku w zakresie „kwestii ludzkich”, który zespoły mogą wykorzystać.

Dlatego ja wierzę, że feedback należy wykorzystywać nie tylko w praktykach technicznych i procesowych, ale przede wszystkim do uzdrawiania komunikacji i relacji w zespole oraz rozwiązywania konfliktów. Czyli po polsku – Jaś mówi Marysi, kiedy Jasia coś w jej zachowaniu irytuje. Problem z feedbackiem jest taki, że jego koncepcja wydaje się prosta jak budowa cepa. W konsekwencji w wielu zespołach stosuje się jego toksyczną wersję „Twój kod śmierdzi” albo stosuje się z rzadka „raz w roku dajemy sobie 'feedback', bo się ocena okresowa zbliża”. Moja rada dla zespołów – „nie idźcie tą drogą, Sabo”.

CZYM FEEDBACK NIE JEST?

Wyznaję zasadę, że najlepiej zacząć naukę od antywzorców, pokazując, jak robić nie należy. Poniżej znajdziesz kilka przykładów pokazujących, jak feedback wyglądać nie powinien:

Byłbyś efektywniejszy, gdybyś stosował test driven development.

Najczęściej spotykany przeze mnie antywzorzec – to nie feedback pane Havranek. Na powyższy zwrot w polskim języku istnieje określenie i nazywa się „rada”.

Dobra robota – świetna prezentacja.

To również nie jest feedback – powyższe zwie się „pochwała”...

Jesteś świnią.

Mam nadzieję, że to tylko mój wymysł i u nikogo w biurze takie słowa nie padły (chyba że jako żart). W barze za takie określenie można oberwać, a w pracy wylądować w HRach. Dlaczego? Bo to nie feedback, ale obelga.

Podobnie jak w przypadku kodu, feedback również można zrefaktoryzować. Oba zadania da się zrefaktoryzować w taki sposób, że będą one bardziej efektywne i przydatne dla drugiej osoby – o tym więcej w dalszej części artykułu.

CZYM JEST FEEDBACK?

Najprostszy feedback to moja zwykła obserwacja na temat drugiego człowieka, np. *Asiu. Masz rozczochrane włosy.* Koniec. Pomocne, ponieważ Asia mogła

nie zauważyć, że na spotkanie z klientem idzie, wyglądając jak Einstein, co klient niekoniecznie uzna za profesjonalne. Zakładając, że Asia wie, że za 15 minut ma spotkanie, oraz rozumie, czemu jej to powiedziałem, i może podjąć natychmiastowe kroki (do łazienki), żeby sprawę wyprostować. Właśnie z tego powodu użycie zdania w formie obserwacji określa się jako „feedback”. Czasownik „feed back” po angielsku jest analogiczny do „give back” („zwrócić coś komuś”), „say back” (odpowiedzieć coś komuś). „We feed someone back”, czyli „karmimy kogoś” (informacją zwrotną na temat czegoś). Stąd też po polsku „feedback” to „informacja zwrotna”.

Przy wykorzystaniu feedbacku powinniśmy pamiętać, że druga osoba, mówiąc kolokwialnie, „nie czyta Ci w myślach”. Jeśli jej nie powiesz, nie będzie wiedzieć, co w danym momencie masz na myśli. Każdy z nas żyje we własnej radosnej rzeczywistości, a to, co się dokoła niej/niego dzieje, interpretuje tak, jak się jej/jemu podoba. Istnieją dwa sposoby obserwacji tej wirtualnej rzeczywistości:

- » zewnętrzna – obserwacja drugiego człowieka,
- » wewnętrzna – obserwacja własnych emocji

Ułatwię Ci lepsze zrozumienie, jak te dwie wirtualne rzeczywistości funkcjonują, prezentując modele wnioskowania, którymi się posługujemy. Do opisanego pierwszej grupy użyję modelu „drabiny wnioskowania” (ang. *ladder of inference*).

MODEL „DRABINY WNISKOWANIA”

Programiści nie są jedynymi, którzy lubują się w modelach. Chris Argyris – pionier badań nad uczeniem się organizacji – opracował model, który ilustruje, jak każdy z nas przechodzi od obserwacji rzeczywistości do podjęcia akcji. Model został nazwany „drabiną wnioskowania”. Przedstawia proces myślowy, który zachodzi u każdego człowieka pomiędzy dokonaniem obserwacji a wykonaniem akcji. Skupia się na tym, jak każdy człowiek subiektywnie odbiera rzeczywistość na bazie przekonania, którymi kieruje się w życiu. Dla ułatwienia zrozumienia modelu „drabiny wnioskowania” przedstawię Wam pewną historię.

Stefan jest starszym programistą w firmie i właśnie w tym momencie krzyczy na młodszego programistę, że zepsuł builda. Stefan najpierw spojrzął na laptopa i zobaczył przeglądarkę, pasek Windows, kilka zakładek, a przede wszystkim stronę z najnowszym rezultatem builda z Jenkinsa (**rzeczywistość**). Na tej stronie zauważył, że build skończył się 10 minut temu i zakończył się niepowodzeniem – sugeruje mu to czerwona ikona kółka (**wybrane elementy rzeczywistości**). Przy wynikach builda widnieje imię i nazwisko młodego programisty Dawida, a ten wiele razy w ciągu ostatnich dni psuł builda, commitując nieudane testy (**wybrane elementy rzeczywistości**). Stefan założył, że wynik ostatniego builda wskazuje na to, kto go zepsuł (**założenia**). Doszedł zatem do wniosku, że to właśnie Dawid niefrasobliwie zepsuł po raz kolejny builda (**wnioski**). Stefan wierzy, że każdy programista, przed commitem, powinien puścić testy, by Jenkins pozostawał zielony (**przekonania**). Postanowił więc, że miarka się przebrała, i na Dawida należy nakrzyczeć (**akcje**), żeby się w końcu ogarnął. Stefan nie wiedział jednak, że tym razem popsuty build nie był winą Dawida. To Jerzy bowiem zmienił konfigurację bazy danych i „wysypał” testy integracyjne. Kiedy Jerzy wychylił się zza monitora i powiedział o tym Stefanowi, ten ze wstydu zrobił się czerwony i błyskawicznie przeprosił Dawida.



Rysunek 1. Model drabiny wnioskowania

Na podstawie modelu „drabiny wnioskowania” widać, że nasze działania będą różne w zależności od dostępnych informacji, przyjętych założeń czy poglądów. Bez poinformowania drugiej osoby o kontekście, albo uzyskania kontekstu od niej możemy pomylić się tak samo jak Stefan. Dlatego przy feedbacku ważne jest, żeby wprowadzić drugą osobę w „nasze realia” i wytłumaczyć, skąd bierze się nasz punkt widzenia. Podobnie ma to miejsce w przypadku, kiedy emocjonalnie reagujemy na sytuację i chcemy naszemu rozmówcy pomóc zrozumieć, dlaczego nagle zachowujemy się irracjonalnie. Taką sytuację lepiej opisać, używając modelu interakcji Satir.

MODEL INTERAKCJI SATIR

Założmy, że obserwujemy rozmowę, która dzieje się kilka dni po incydencie opisanym powyżej. W tym czasie Dawid „wysypał” jeszcze kilka buildów i postanowił porozmawiać na ten temat ze Stefanem. Rozmowa wygląda w następujący sposób:

Dawid: *Obiecuję, że tym razem się poprawię i już za każdym razem będę puszczał testy.*

Stefan: *W porządku.*

Dawid: *Dlaczego mi nie ufasz?*

Łatwo mi wyobrazić sobie zdumienie pojawiające się na twarzy Stefana po słowach Dawida. Jestem pewien, że podobnie jak on zastanawiasz się, w jaki sposób Dawid wywnioskował, że jego kierownik projektu mu nie ufa. Zrozumienie tej zagadki ułatwia przeanalizowanie odpowiedzi Dawida za pomocą modelu interakcji Satir. Dawid usłyszał „w porządku” (**obserwacja**) – mógłby założyć, że Stefan przyjął do wiadomości, zaakceptował przeprosiny i uwierzył, że Dawid rzeczywiście się poprawi. Dawid nie jest natomiast naiwniakiem i zdaje sobie sprawę, że te same słowa mogą mieć wiele interpretacji. Dawid nie ma dobrego doświadczenia z podobnymi sytuacjami i na tej bazie interpretuje odpowiedź Stefana jako lakoniczną (**interpretacja**), znaczącą *Jestem pewien, Dawid, nieraz będę musiał z Tobą na ten temat rozmawiać*. Dawid czuje, że krótka odpowiedź Stefana jest w istocie ironicznym komentarzem (**emocja**). Czuje się urażony taką odpowiedzią (**emocja w stosunku do emocji**) i decyduje bronić się przed ironicznym komentarzem (**obrona**). W przypadku Dawida obrona opiera się o potrzebę wyrażaną jako „Chcę wiedzieć, dlaczego ludzie uważają, że nie potrafię czegoś zrobić” (**reguła do reakcji**). W rezultacie zaaplikowania tego wewnętrznego procesu pojawia się z ust Dawida pytanie *Dlaczego mi nie ufasz?* (**wynik**).

obserwacja
interpretacja
emocja
emocja w stosunku do emocji
obrona
reguła do reakcji
wynik

Rysunek 2. Model komunikacji Satir

Będąc w opisanej sytuacji (niezależnie, po której stronie się znajdziesz), powinnaś/powinieneś pamiętać, że feedback jest subiektywny i może powstawać na bazie reakcji emocjonalnej. Reakcja emocjonalna to wyłącznie „wierzchołek góry lodowej”¹. Pod jej powierzchnią znajdują się potrzeby drugiej osoby, jej wartości, wizerunek siebie, ukryte oczekiwania itd. Warto pamiętać, że tego typu sytuacje są nieuniknione. Za tę reakcję odpowiada fragment mózgu starszy ewolucyjnie, nad którym mamy ograniczoną kontrolę. W celu uzdrowienia sytuacji i przekazania właściwego feedbacku należy przeprowa-

¹ nomen omen, model komunikacji Satir opiera się o model osobowości nazywany modelem góry lodowej

dzić racjonalną rozmowę. Racjonalna rozmowa pozwoli zrozumieć emocje, zidentyfikować problem i lepiej komunikować się w przyszłości.

REFAKTORYZACJA DO FEEDBACKU

Model „drabiny wnioskowania” oraz model interakcji Satir wyjaśniają, w jaki sposób powstają konflikty w interakcjach międzyludzkich. Odkrywają dokładne mechanizmy, które doprowadzają do ich powstawania. Znając te mechanizmy, możesz zapobiec powstawaniu konfliktów i efektywniej formułować feedback. Pokażę to na przykładzie „refaktoryzacji” przykładów kiepskiego feedbacku.

Pisałbyś lepszej jakości kod, gdybyś stosował Test Driven Development.

Łatwo zauważyć, że osoba wypowiadająca te słowa kieruje się przyjętym **przekonaniem**. Przyjmuje, że Test Driven Development jest sposobem efektywnego tworzenia oprogramowania, a co najmniej, że jest efektywniejszy niż metoda, którą druga osoba stosuje teraz. Nie kwestionuję tego założenia, ale stwierdzenie „byłbyś efektywniejszy” to przepowiadanie przyszłości i jest na 100% pewien, że to akurat jest niemożliwe.

W feedbacku przyjęto też **założenie**, że drugiej osobie zależy na dobrej jakości kodu. Jeśli nie jest to prawda, feedback rozwiązuje problem, którego nie ma i jest mało przydatny.

Feedback jakościowo lepszy mógłby brzmieć:

Na bazie moich doświadczeń Test Last Development jest mniej efektywny od Test Driven Development. Zakładam, że jakość kodu jest dla Ciebie ważna. Myślę, że mógłbyś pisać lepszej jakości kod, gdybyś stosował TDD.

W pierwszym zdaniu osoba przekazująca feedback tłumaczy, jakie przyjęła założenie i dlaczego. W drugim zdaniu wyjaśnia kontekst, bez którego feedback może nie mieć sensu. Oba zdania wskazują na empatię osoby mówiącej, i wyrażają chęć zrozumienia. I dopiero w zdaniu trzecim pojawia się „rada”.

Dobra robota – świetna prezentacja.

Powyższa wypowiedź nie zawiera żadnych konkretnych informacji. Przyjmijmy więc, że osoba wypowiadająca to team leader. Mogło nią/nim kierować przekonanie, że kolegę z zespołu należy pochwalić, kiedy wykazuje inicjatywę. Sama prezentacja mogła jej/jemu się nie bardzo podobać i być wprost nudna. Dając feedback w takiej formie jak powyżej, team leader wyrządza pracownikowi niedźwiedzią przysługę. Ten może uznać, że było dobrze i zdecydować przedstawić podobną prezentację w szerszym gronie – gdzie uczestnicy nie wybaczą jej/jemu straconego czasu tak łatwo. W podobnej sytuacji team leader powinien wprost powiedzieć:

„Podoba mi się, że wyszedłeś z inicjatywą tej prezentacji – myślę, że to ciekawy temat i przydatny w naszym zespole. Z mojej perspektywy poruszyłeś go bardzo abstrakcyjnie – brakowało mi historii i przykładów, przez co trudno było mi utrzymać uwagę”.

Nie ma tu owijania w bawełnę, a jednocześnie feedback nie „miażdży” drugiej osoby. Ta wypowiedź jest szczerą, podaje czytelnie **obserwacje** oraz komunikuje **przekonania** team leadera.

Jesteś świnią.

W tym przypadku mamy do czynienia z emocjonalną reakcją na słowa, bądź zachowanie drugiej osoby. Nie przeprowadzę refaktoryzacji tej wypowiedzi, a zamiast tego zajmę się refaktoryzacją sytuacji. Zadaję sobie pytanie: „Jak mogłbym zareagować, żeby wyprowadzić tę rozmowę na prostą”? Dużym błędem byłoby odniesienie się bezpośrednio do słów „jesteś świnią”. Z modelu interakcji Satir wynika jasno, że dużo bardziej istotne jest to, co kryje się „pod powierzchnią”. W tym przypadku wyciągnąłbym rękę do osoby mówiącej, pokazując pozytywne intencje. Sięgnąłbym głębiej, odczytując towarzyszące temu zdaniu emocje: *jesteś wyraźnie zdenerwowana/zdenerwowany tym, co powiedziałem*. Następnie zadeklarowałbym chęć zrozumienia, stwierdzając na przykład: *czy uraził Cię mój komentarz na temat Twojego koloru włosów?* W ten sposób dążę do tego, aby prowadzić dyskusję na wszystkich poziomach modelu Satir. Sięgam do źródła problemu – zamiast skupiać się na niewiele znaczącej reakcji emocjonalnej².

RADY PRAKTYCZNE

1. Upewnij się, że praktyka feedbacku, model „drabiny wnioskowania” oraz model komunikacji Satir jest znany w Twoim zespole oraz przez niego zaakceptowany,
2. Skup się na obserwacjach oraz na tym, w jaki sposób zachowanie innych wpływa na Ciebie. Używając modelu „drabiny wnioskowania”, wytłumacz swój tok myślenia drugiej osobie.
3. Pamiętaj, że wielu sytuacjom towarzyszą emocje, nad którymi mamy małą kontrolę (słuchanie feedbacku jest często sytuacją stresową!). Zastosuj model interakcji Satir, aby zrozumieć emocje i przejść do racjonalnej komunikacji.
4. Dawaj feedback często, aby zapobiec tworzeniu się toksycznego środowiska pracy.
5. Zapoznaj się z różnymi technikami feedbacku, np. feedback 360.

PODSUMOWANIE

Feedback jest jedną z podstawowych technik, które pomagają utrzymaniu zdrowych relacji w zespole. W artykule pokazałem, jak użyć dwóch modeli komunikacji do przekazywania go efektywnie. Ich esencją jest krytyczna analiza procesu myślenia i przekazanie swoich racji w sposób pełny oraz z uwzględnieniem perspektyw drugiej osoby. Kluczem do uzyskania maksymalnych korzyści z feedbacku jest dobra znajomość tych modeli w zespole oraz częste ich wykorzystanie w codziennej pracy. Pomocne jest przeprowadzanie ćwiczeń refaktoryzacji feedbacku samemu lub, jeśli to możliwe – w większych grupach.

Wprowadzenie feedbacku jest pierwszym krokiem do wdrożenia w zespole techniki mikro-coachingu. Więcej o mikro-coachingu oraz o tym, jak skutecznie implementować feedback, opowiem w kolejnych odsłonach serii.

² Algorytm zdrowej komunikacji w podobnych sytuacjach opiszę dużo dokładniej w kolejnych odsłonach serii, kiedy przedstawię model „komunikacji bez przemocy” Rosenberga.

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.

