

DARIUSZ JEMIELNIAK

*Wyższa Szkoła Przedsiębiorczości i Zarządzania
im. Leona Koźmińskiego, Warszawa*

Menedżerowie w oczach informatyków¹

Tako rzecze mistrz programowania: Niechaj programistów będzie wielu, a menedżerów mało – wtedy wszyscy będą produktywni
(James, 1986)

1. Geneza konfliktu specjalista–menedżer

Jak twierdzi Schein (1991) poszczególne jednostki (np. założyciele firm, dyrektorzy, a więc przede wszystkim osoby pełniące funkcje menedżerskie) mogą wywierać dominujący wpływ na kształt relacji i kulturę organizacji. Jednakże, jak wskazuje wielu autorów (por. Rosen, 1991; Kunda i Van Maanen, 1999) kultura organizacji z perspektywy menedżerskiej może różnić się znacznie od postrzegania jej przez pracowników, a w szczególności różne grupy zawodowe. Co więcej, kultury zawodowe często mogą kształtować się w opozycji do kultury menedżerów i ich perspektywy kultury organizacyjnej. Zarządzanie zatem niemal zawsze ma charakter międzykulturowy (Koźmiński, 2004).

¹ Alternatywna wersja interpretacji prezentowanych tutaj badań ukazała się w lipcu 2007 na łamach *Journal of Organization and Change Management*.

Chociaż wielu teoretyków twierdzi, że „menedżerowie i profesjonalści (a zwłaszcza inżynierowie) są tymi, którzy najbardziej identyfikują się z firmami, dla których pracują” (Kunda i Van Maanen, 1999: 64), i w związku z tym mogliby być postrzegani jako naturalni sojusznicy, wielu badaczy zauważa jednocześnie, że „menedżerowie uczą się wykorzystywać do ostatka inżynierów i grupy projektantów” (Kunda, 1992: 44). Nie jest więc zaskakujące, że w wielu firmach zaobserwować można pogłębiający się konflikt właśnie między menedżerami a specjalistami (Hall, 1986; Trice, 1993).

Według Scheina (1985) istotną własnością przywództwa jest zdolność manipulowania kulturą, wpływania na symbole, normy, wartości i założenia. Choć kultury z trudem poddają się prostej manipulacji, zgadzam się, że charakterystyczną cechą kultur profesjonalnych jest dążenie do wyzwolenia spod kontroli menedżerskiej i zapobieganie rutynie (Trice, 1993). Ciekawe wydaje się zatem zbadanie tego konfliktu w przypadku programistów, których zawód w szczególności opiera się na unikalnej wiedzy.

W wielu współczesnych organizacjach podejmuje się próby łagodzenia opozycji menedżerowie–specjaliści (Kunda, 1992). Np. w IBM (w książce Kundy występującym pod pseudonimem High Technologies Corporation) inżynierowie–programiści podczas różnych szkoleń byli usilnie przekonywani, że unikalna kultura firmy jest szczególnie nastawiona na pracowników, a wśród jej licznych zalet istotnie wyróżnia się brak podziału na menedżerów i inżynierów, zaufanie dla kompetencji profesjonalistów, brak przepaści między osobami o różnym autorytecie formalnym, zanik tradycyjnej struktury, podejmowanie decyzji na poziomie późniejszych realizatorów itp. Jednak, jak zauważają Kunda i Van Maanen (1999), jest to często gra pozorów – obietnice braku zwolnień okazują się być bez pokrycia, a – choć faktycznie wiele korporacji wprowadza procedury zwiększające kreatywność, komfort pracy itp. – to w naprawdę istotnych kwestiach, jak restrukturyzacja, *out-sourcing*, zamykanie fabryk, zwolnienia, nie ma mowy o żadnej równości i rolę odgrywa tradycyjna hierarchia.

Programiści są szczególnie interesującą grupą zawodową także dlatego, że reprezentują bardzo silną (kontr)kulturę (Trice and Beyer, 1993; Kunda and Van Maanen, 1999; Barley i Kunda, 2004), a jednocześnie są sztandarym przykładem „pracowników wiedzy”, pracujących pod niezwykłą presją i bardzo stresującym środowisku (Kidder, 1981; Gephart, 2002), często w organizacjach posługujących się *kontrolą normatywną* (Kunda, 1992). Nie

może to pozostawać bez wpływu na ich relacje z kierownictwem – rzadko jest jednak przedmiotem analizy.

W tym świetle warto zastanowić się, jak programiści postrzegają menedżerów, jaki jest ich stosunek do organizacji, w których pracują, jak postrzegają swój własny zawód i karierę, wreszcie: w jaki sposób sprawowana jest kontrola nad produktem (która z kultur dominuje wg typologii Trice’a i Bayer, 1993) i w jaki sposób rozwiązywany jest konflikt między tymi dwiema grupami, jeżeli do niego dochodzi.

2. Metoda badań

Niniejszy tekst opiera się na etnograficznym studium trzech polskich przedsiębiorstw z branży IT. Wykorzystane metody obejmują otwarte, mało ukierunkowane wywiady (43 programistów i 4 menedżerów), *shadowing*, a także *storytelling*, należą więc do metod jakościowych, inspirowanych antropologią organizacji i teorią ugruntowaną (Konecki, 2000; Kostera, 2003). Wszystkie rozmowy były nagrywane i transkrybowane za zgodą rozmówców (których prawdziwe imiona zostały usunięte z tekstu w trosce o anonimowość badanych), każdy wywiad trwał 50–70 minut. Rezultaty badań mają charakter performatywny, a nie ostensywny (Latour, 1986; Jemielniak, 2002): celem analizy jest zaprezentowanie punktu widzenia badanej grupy, a nie stworzenie teoretycznego modelu, a autor przyjmuje nieuprzywilejowaną rolę badacza (Greenwood i Levin, 1998). Dane poddane analizie są z pewnością częściowo wynikiem zbiorowych stereotypów. Stereotypy są jednak także użytecznym przedmiotem badań – to za ich pomocą powiela się kultura zawodowa, stanowią istotny przejaw nadawania sensu rzeczywistości organizacyjnej (Boje, 1991), konstruowanej w interakcji (Weick, 1969/79). Dlatego celem analizy nie jest ustalenie „jak jest naprawdę”, a prezentacja obrazu, jaki rozmówcy kreślą i kreują na potrzeby wywiadu².

² Mimo że jest to specyficzny rytuał interakcyjny (Goffman, 1967), przyjmowana w nim poza wynika z tego, co środowisko programistów uważa za możliwe do przyjęcia, nadające się do przekazania w rozmowie – co samo w sobie jest przejawem określonej kultury tej zbiorowości.

3. Menedżerowie i kariera

Punktem wyjścia do dalszych rozważań będzie analiza ich stosunku do menedżerów. Z przeprowadzonych badań wynika, że jest on zdecydowanie negatywny. Menedżer często jest zdefiniowany jako osoba, której brakuje wiedzy. Co więcej, stereotypowy kierownik wcale nie ukrywa swojej ignorancji – przeciwnie, próbuje uchodzić za posiadającego odpowiednią wiedzę, być może wręcz nie zdając sobie sprawy z własnych niedostatków (Wodan4):

[P:] A dlaczego? To jest ciekawe.

[O:] No bo przede wszystkim... Znaczy, jak wygląda proces rekrutacji na menedżerów? Jak wyglądają te studia menedżerskie? Są to studia, które naprawdę są to studia nad wskaźnikami i jak je wałkować. Znowu mówię bardzo generalnie, bo na pewno się zdarzają świetni menedżerowie, którzy są utalentowani, ale generalnie... I menedżerem tak naprawdę, mam wrażenie, zostaje człowiek, który nie bardzo wie, kim chce zostać. A... Bardzo często u menedżerów jest tak, że mają wymagania niemożliwe do zrealizowania. To znaczy na przykład, jeżeli się mówi, żeby napisać superprogram w ciągu dwóch dni. Jest to niemożliwe, po prostu – fizycznie niemożliwe. Nawet gdyby ktoś siedział te czterdzieści osiem godzin, to by nie napisał, bo nie ma takiej możliwości. Inne rady dla menedżerów... nie mam pojęcia. Nie czuję się jakoś kompetentny do dawania rad menedżerom. Wszystko, tyle że, jeżeli chcą... Jeżeli chce się rzeczywiście jakiegoś takiego efektu, to trzeba zainwestować w tego, kto jest menedżerowany.

Programista neguje nie tylko wiedzę menedżera na temat realiów realizacji projektów informatycznych, ale samą naukę o zarządzaniu jako taką. Studia z zakresu zarządzania przedstawione zostają jako „wałkowanie wskaźników”. Ponadto programista zwraca uwagę, że osoby, które decydują się na karierę menedżerską, wybierają ją, bo nie potrafią robić nic innego, jest to więc selekcja negatywna. Wreszcie, zdaniem mojego rozmówcy typowy menedżer charakteryzuje się nierealnymi żądaniem.

Nic dziwnego, że osoby, które były programistami, a teraz zajmują się zadaniami menedżerskimi albo po prostu nieprogramistycznymi postrzegane są przez innych programistów jako przegrani. Zupełnie jak w opisie Ullman (1996), przedstawiającym programistę, który nie pisał wystarczająco dobrze:

Jego program nie spełniał wszystkich funkcji, które powinien i ukarano go za to. Jego karą była konieczność kontaktowania się ze zwykłymi ludźmi. Frank został inżynierem do spraw sprzedaży. Co zabawne,

pracując w dziale sprzedaży i mając udział w premiach, zarabiał więcej. Ale nie miał już opcji na akcje. A w oczach innych inżynierów Frank był tak „wysoki”³ jak tylko się dało. Gdy nas pytano, mówiliśmy „Frank zajmuje się teraz sprzedażą”. Równie dobrze mogliśmy mówić, że nie żyje.

Pod wpływem tego silnie negatywnego obrazu zarządzania znajdowała się także percepcja kariery moich rozmówców (Minicorp5):

[P:] A jakie byłoby stanowisko, na którym Pan by najbardziej chciał pracować, gdyby Pan był nie wiem, dajmy na to... Za dziesięć lat, jak Panu super pójdzie w karierze...

[O:] Jeszcze pytanie, czy ja w ogóle chcę zrobić karierę.

[P:] No właśnie, jak Pan to widzi?

[O:] Maksymalne stanowisko, jakie widzę dla siebie to jest lider projektu, nie wyżej.

[P:] Dlaczego?

[O] Bo lubię programować i mieć związek z tworzeniem oprogramowania, natomiast jakby już kwestie kontaktu z klientami, niezupełnie mnie interesują.

Mój rozmówca nie tylko nie chciał sprawować funkcji menedżerskich. Na pytanie o „karierę” wyraźnie zaznaczył, że to pojęcie nie jest dla niego istotną kategorią definiującą. Podkreślił także, że najbardziej interesuje go samo programowanie. Odrzucenie w ogóle pojęcia „robienia kariery”, stanowiącego w moim przekonaniu jeden z istotnych fundamentów menedżerskiego języka i światopoglądu, na różne sposoby było sygnalizowane przez wielu z moich rozmówców. Zdaniem Halla (1986: 49):

[Z] askakująco duża liczba ludzi zmienia zawód z profesji, w której zaczęli swoją karierę (...). Wiele z tego ruchu ma miejsce w profesjach inżynierskich. Wynika to z tego, że ludzie awansują ze stanowisk inżynierskich na menedżerskie (inżynierowie mogą być też odsunięci na boczny tor lub wyrzuceni z pracy...).

Nie znajdowało to jednak potwierdzenia w wypowiedziach moich rozmówców. Niektórzy z nich np. w ogóle kwestionowali sensowność pytań o to, co się będzie robiło w dalekiej przyszłości (Minicorp9):

³ Ullman opisywała podział na języki programowania wysokiego i niskiego rzędu – im lepiej programista był w stanie pisać językiem niskiego rzędu (zbliżonym do maszynowego), tym wyższy status zajmował w hierarchii. Potocznie programiści przymiotnikiem „wysoki” określali osoby programujące słabo lub wcale.

[P:] A kim by Pan chciał być za jakieś dziesięć lat?

[O:] Wie Pan co? Ja powiem Panu coś takiego – ja kiedyś poszedłem na rozmowę kwalifikacyjną do jednego banku, zresztą polecony byłem przez mojego kolegę i przyszła taka bardzo młoda pani z działu *human resources* chyba i spytała mnie: „Kim chciałby Pan być?”. Spytała mnie tam, wie Pan, w ciągu tam pierwszych trzech pytań. I siedział obok szef działu IT, który jak to usłyszał to spojrzął na nią takim bardzo ciekawym wzrokiem. No to ja jej odpowiedziałem, że prezesem bym chciał być. Na co ona już mnie więcej na rozmowę nie zaprosiła. To pytanie było kretyńskie, po prostu, wie Pan, kim ja bym chciał być w ciągu następnych dziesięciu lat. I wie Pan, za dziesięć lat to... No ona nie doceniła, że chciałem być prezesem jej banku, no wie Pan, za dziesięć lat. Ja na razie mam cel prosty – ja chcę zrobić doktorat, tutaj na razie w tej firmie jest mi dobrze, także, wie Pan, jeśli ja za dziesięć lat będę miał doktorat i ta firma będzie może się rozwijała, ale ja zachowam jakby tą samą pozycję w tej firmie, to ja będę bardzo zadowolony.

Mój rozmówca w delikatny sposób dał mi do zrozumienia, że moje pytanie przypominało „kretyńskie” pytanie pracownicy działu HRM. Owa osoba, nawiasem mówiąc, została określona jako „bardzo młoda pani” (co może sugerować brak wiedzy i doświadczenia). Programista podkreślił także wspólnotę percepcji z szefem działu IT (który „spojrzął na nią bardzo ciekawym wzrokiem”). Pytanie o karierę i wizję siebie za dziesięć lat było dla niego tak absurdalne, że podczas rozmowy kwalifikacyjnej zareagował na nie prowokacyjnym żartem. Ta bardzo ciekawa opowieść została podsumowana zanegowaniem chęci kariery – mój interlokutor zauważył, że będzie bardzo zadowolony, jeżeli w przyszłości po prostu zachowa tę samą pozycję, którą ma obecnie. W podobny sposób wypowiadali się inni programiści (Sand11):

P: Jak Pan widzi swoją przyszłość zawodową?

O: Znaczy wie Pan, no mogę jakby awansować, ale od razu powiem, to nie jest moim jakimś... Ja lubię programować, nie widzę najmniejszego powodu, dla którego ja w momencie, kiedy bym miał zacząć w jakiś sposób realizować takie funkcje zarządzania, miałbym odewracać się od technicznej strony, to jest wręcz przeciwnie. Wydaje mi się, że żeby dobrze kierować pracą zespołu, należy wiedzieć dość dokładnie, co ten zespół robi. Oczywiście wtedy się człowiek częściowo wyłącza z takich zadań projektowo-programistycznych, ale nie sądzę, żebym to w 100% chciał kiedyś się z tego wyłączyć. Bo to ani mi nie pomoże, ani nie będzie dla mnie wcale przyjemne.

Kwestie awansu, uzyskiwania wyższej pozycji w hierarchii organizacyjnej miały dla programistów znikome znaczenie. Wszyscy wyraźnie podkre-

ślali, że pisanie programów sprawia im na tyle dużą przyjemność, że nie chcą z niej rezygnować na rzecz organizacyjnych wyróżnień. W odmiennym tonie wypowiedzieli się natomiast o sprawowaniu kontroli nad pracą innych, kierowaniu – wyrażali przekonanie, że gdyby już to mieli robić, sprawowanie władzy nie byłoby dla nich niczym przyjemnym. Nawet właściciel Minicorp, były programista, stwierdził, że doskonale rozumie niechęć programistów do sprawowania funkcji menedżerskich:

P: Jak Pan sądzi, czy programiści z Pana firmy chcieliby być w przyszłości menedżerami, jak Pan?

O: Ja myślę, że na pewno nie i bardzo dobrze wiem dlaczego. Dlatego, że sam też nie byłem zachwycony tym, że w tę stronę rozwija się firma, znaczy no, miałem tego świadomość, ale to jest zupełnie inna praca. Bo praca programisty jest pracą twórczą – tworzy coś, człowiek tworzy, jest to praca ciekawa i jeśli ktoś to lubi, to wydaje mi się, że nie chce uciekać od programowania, bo jest to znacznie ciekawsze niż zarządzanie ludźmi dla takiej osoby.

P: A dla Pana też?

O: Ja myślę, że tak. To znaczy bardzo lubiłbym być programistą i to mi sprawiało myślę największą satysfakcji w mojej pracy. Ale nie na tyle dużo, żeby zostać na etapie programisty.

Nawet właściciel firmy software'owej wyznał, że praca programisty sprawiała mu największej przyjemności zawodowej. Pisanie programów zostało określone jako praca twórcza (ten wątek zostanie rozwinięty w późniejszych rozdziałach), z której niechętnie rezygnuje się na rzecz innych działań. Zwłaszcza że sama praca menedżerska kojarzona jest z nudą i bezcelowymi procedurami. Chęć awansu wyrażało zaledwie kilku rozmówców, od razu zastrzegając, że chcieliby pozostać na stanowisku związanym z programowaniem (Wodan11):

P: No a tak patrząc, za dziesięć lat, na jakim stanowisku by Pan w idealnej sytuacji sukcesu chciał być?

O: To znaczy, powiem szczerze, że chciałbym pozostać jednak, na pewno wyższą funkcję jakąś zarządzająco... Ale jeszcze taką, która miałaby dużo *à propos* technologii. Czyli od programisty może przejść do analityka...

P: A czemu?

O: Czemu... To jest strasznie interesujące. Bo analityk analizuje potrzeby klienta, zna technologię i jakby kreuje wizję systemu. Że: „no, słuchajcie chłopaki, ten system będzie oparty o takie technologie, ponieważ to, to i to”. Dla analityka, pewnie też się to może znużyć po jakimś czasie i może się okazać, że już. Jeszcze się nie zastanawia-

łem, czy chciałbym być szefem jakiegoś dużego działu, bo to jest jednak biurokracja. Ja nie wiem, czy chciałbym się zajmować biurokacją. No może to jest jednak tak: papierki, tysiące maili. Natomiast... ja, mnie interesuje taka praca... Na przykład, ja chętnie bym się widział na stanowisku, gdyby, powiedzmy, możliwości finansowe na to pozwalały, to mógłbym w jakimś zespole badawczym pracować. Zresztą każdy informatyk, programista Panu powie: tworzyć nowe rzeczy, rozwiązywać jakieś niestandardowe problemy. To jest interesujące, to jest to, co jest ciekawe.

Podstawowym kryterium atrakcyjności pracy była jej niestandardowość, zdolność do wzbudzania zainteresowania. Praca menedżerska kojarzona była natomiast z biurokacją w najgorszym sensie tego słowa. Jak mówił jeden z programistów, jedyny, który rozważał karierę menedżerską (Minicorp1):

P: Jak postrzegasz swoją pracę w przyszłości?

O: No, ja to właściwie idę tak z *flow* w tej chwili. Jeżeli tak dalej pójdzie to pewnie zostanę menedżerem, bo w tej chwili nie wydaje mi się już to takie straszne jak kiedyś.

P: A kiedyś Ci się wydawało?

O: Tak, parę lat temu jeszcze mi się wydawało, a w tej chwili mi się nie wydaje.

P: A dlaczego Ci się wydawało?

O: No bo kiedyś, że tak powiem, kulturowo wydaje mi się, że zaliczałem się do tych kilku, kilkunastu procent, ale dokonałem przejścia kulturowego.

P: Możesz mi wyjaśnić to, bo ja nie wiem, na czym polega ta różnica kulturowa?

O: No istnieje taki podgatunek informatyków, wydaje mi się, którzy są dumni głównie ze swoich umiejętności nazwijmy to technicznych, inżynierskich i wśród tego gatunku właśnie znajdują się ludzie, którzy nie lubią menedżerów czy też czują jakiś *culture crash* z kulturą menedżerską i nie wyobrażają sobie siebie w roli przyszłych menedżerów. Zazwyczaj droga kariery takich ludzi jak oni sobie ją wyobrażają, to jest w którymś momencie przejście z pozycji zatrudnionego w jakiejś takiej firmie na założenie na przykład własnej firmy zajmującej się takim konsultingiem technicznym czy też własnej firmy piszącej oprogramowanie, zwłaszcza jakieś wyrafinowane technicznie oprogramowanie dla innych programistów – tego typu zadania.

Mój rozmówca, w przeciwieństwie do pozostałych, posługiwał się słownictwem wskazującym na znajomość literatury menedżerskiej lub socjologicznej („culture clash”, „kultura menedżerska”). Jako jedyny wprost przed-

stawił konflikt między menedżerami a programistami jako wynik pewnych odmienności kultur zawodowych. Wskazał też na wymarzony model kariery dla programisty (który potwierdziło także wielu badanych) – założenie własnej firmy lub konsulting. Dla obydwu tych wariantów charakterystyczne jest wyzwolenie się spod władzy menedżerów i organizacji, przy jednoczesnym pozostaniu przy programowaniu.

Praca w dużej firmie wielu programistom kojarzy się z koniecznością uczestnictwa w walkach organizacyjnych, „wygryzaniu innych”, czego nie chcą lub nie potrafią robić. Obawa przed niekompetentnymi zwierzchnikami, częściej występującymi w dużych przedsiębiorstwach, była powszechna (Wodan4):

[P:] A dlaczego nie chciałby Pan pracować w większej firmie?

[Q:] To jest trochę tak, że można spędzić całe życie w korporacji i nawet nieźle awansować, ale cały czas robić jakieś bzdety. Albo, dajmy na to, harować jak wół, powiedzmy przez tydzień, żeby się dowiedzieć, że zmieniła się koncepcja i robota jest do kosza. Mój znajomy tak miał. Więc niektórzy to mogą lubić, ale praca tego typu to jest nie dla mnie.

Wszystkie przytoczone wypowiedzi świadczą o tym, że programiści, z którymi rozmawiałem, mają bardzo złe zdanie o dużych firmach. Nawet moi rozmówcy z Sand, czyli jednej z największych polskich firm software'owych, podkreślali, że ich firma stanowi wyjątek od stereotypu, ale samego stereotypu nie kwestionowali.

Mój rozmówca czuł się w obowiązku wytłumaczyć z pracy w dużej firmie, podkreślał, że nie charakteryzuje się ona domniemanymi wadami. Wyraźnie też zaznaczał, że dział, w którym pracuje, jest strukturalnie oddzielony od całej organizacji i w związku z tym nie dzieli jej ułomności.

4. Konflikt jako zderzenie kultur

Wszystkie te wypowiedzi świadczą o poważnym antagonizmie między programistami a menedżerami (a także często uosabianymi przez nich organizacjami). Warto w tym miejscu zastanowić się nad wnioskami z niego wypływającymi. Punktem wyjścia może być obserwacja jednego z moich rozmówców (Wodan3):

P: Czym się różnią podejścia do pracy informatyka i menedżera?

O: Generalnie nie jest dobrze... Wiadomo, menedżer zawsze chce szybko zrobić tak, żeby klient był zadowolony. Oczywiście, my też generalnie jesteśmy za tym, tylko ogólnie dla mnie na przykład ważne jest jeszcze, żeby napisać porządny program.

W przekonaniu mojego interlokutora przyczyną częstych rozbieżności między menedżerami a programistami jest przywiązanie do produktu tych ostatnich. Podobnie twierdzi Burkard Sievers (1990: 88): „Menedżerowie zazwyczaj utożsamiają się, a także są utożsamiani przez innych, z zatrudniającą ich instytucją, podczas kiedy sami pracownicy, jako że nie są już w stanie dorównać w doskonałości swoim produktom, zazwyczaj utożsamiają się ze swoimi produktami”.

Antagonizm menedżerowie–programiści może mieć jednak bardziej fundamentalne podłoże. Wypowiedzi moich rozmówców podsumować można następująco:

1. Uważają menedżerów za ogólnie niekompetentnych, choć często przekonanych o swojej wiedzy.

2. W większości odrzucają pojęcie kariery, powszechnie podkreślając przy tym wartość, jaką jest dla nich ciekawa praca. Kilkukrotnie przywoływanym modelem idealnym przyszłego zajęcia jest posiadanie własnej małej firmy lub konsulting – a zatem zajęcia pozwalające na zajmowanie się programowaniem, a przy tym uwalniające od bycia podwładnym.

3. Szczególnie negatywnie odnoszą się do formalizacji (uważają ją za „papierkologię”) i politycznej gry o władzę i zasoby, które są szczególnie charakterystyczne dla dużych firm.

Z powyższych obserwacji można wyprowadzić wniosek, że kultura zawodowa programistów w dużym stopniu jest *de facto* kontrkulturą, czyli kulturą rozwijającą się w opozycji względem innej, dominującej kultury (w tym przypadku menedżerów), tworzy własne wartości. Programiści odrzucają władzę zarówno jako wartość oferowaną przez organizacje (awans, kariera) jak i relację zwierzchnictwa (stosunek przełożony-podwładny). Miejsce władzy zajmują wartości wypływające z wiedzy – interesujące problemy, swoboda decydowania o swojej pracy, możliwość robienia ciekawych rzeczy, zdobywanie nowych umiejętności.

To bardzo rudymetarna dychotomia – przedstawiciele obu tych kultur wyznają różne wartości, różnią ich podstawowe założenia co do celu funkcjonowania w organizacji, nic więc dziwnego, że nierzadko dochodzi między nimi do nieporozumień.

Tabela 7. Model do klasyfikowania relacji między menedżerami a kulturami zawodowymi

		Kontrola według reguł pionu administracji	
		dominuje	jest pomijana
Kontrola według reguł grupy zawodowej	dominuje	wzajemna tolerancja i dopasowanie (np. prawnicy i lekarze w firmach)	dostosowanie menedżerów do grupy zawodowej (np. uniwersytety, szpitale)
	jest pomijana	dostosowanie grupy zawodowej do menedżerów (np. księgowi, inżynierowie, zawody nie wymagające kwalifikacji)	egalitaryzm - traktowanie menedżerów i pracowników w podobny sposób (np. restauracje, alternatywne szkoły)

Opracowanie własne, na podstawie: Trice i Beyer, 1993: 188

Według większości klasyfikacji programistów należy uznać za specjalistów – ich kluczowym zasobem w organizacji jest wiedza. Chociaż są także przedstawicielami zawodu tradycyjnie uważanego za inżynierski (moi rozmówcy także sami niejednokrotnie używali w stosunku do siebie tego terminu), z całą pewnością można stwierdzić, że profesjonalizacja rozumiana jako stopień kontroli nad przebiegiem procesu powstawania produktu, jest w ich przypadku daleko posunięta. To programiści wyznaczają standardy oceny produktu, menedżerowie bez przeszłości w informatyce nie są nawet w stanie ocenić tego, co programiści im komunikują. Jak powiedział jeden z moich rozmówców (Sand 9):

P: Jakie kryteria powinien spełniać dobry kierownik projektu?

O: Mi się wydaje, że to jest raczej bardzo wskazane, żeby był w przeszłości programistą.

P: Tak, a czemu?

O: No bo dzięki temu łatwiej jest zrozumieć to, co... Jakby w pewnym sensie to jest tak, że kierownik projektu powinien zarządzać grupą programistów, którzy ten projekt realizują. Wyobraźmy sobie kierownika projektu, który w ogóle nie ma pojęcia o programowaniu, moim zdaniem to jest kompletna katastrofa. On nie jest w stanie zrozumieć w ogóle problemów, które mogą się pojawić w takim projekcie. Bo oczywiście taki projekt w pewnym sensie jest pracą przy założeniach: my zakładamy, że jesteśmy w stanie zrealizować takie i takie funkcjonalności w takim i takim koszcie, prawda. No i oczywiście wszystko jest świetnie, jak tak się dzieje. A pytanie co się dzieje, jak się pojawiają problemy... I mi się wydaje, że kierownik projektu bez doświadczenia programistycznego to będzie miał trudności ze zrozumieniem jakby skali problemu, prawda? Przykład: programista mówi, że nie jest w stanie tego napisać, no i teraz pytanie jest: czy rzeczywiście programista jest leniwy, czy raczej słaby, czy programista ma rację, bo kierownik projektu źle zrobił projekt?

Dzięki samodzielnemu definiowaniu standardów wykonania pracy, programiści są w stanie jako jedna z nielicznych grup zawodowych definiować swoją kulturę według pojęć spoza tradycyjnego menedżerskiego słownika. Nie respektują rytuałów narzucanych przez kierownictwo – biorąc pod uwagę ich konieczną wiedzę ekspercką i przejmowanie kontroli nad oceną rezultatów pracy, można rozumieć ten proces jako wynik profesjonalizacji zawodowej tej grupy. Dzięki niej programiści mogą odrzucać kierowniczą „propagandę”.

Jak zauważa Rosen (1986/91), używając dokładnie tego samego słowa, w dużych organizacjach często mamy do czynienia właśnie z propagandą kulturową. Rosen tak opisuje przebieg i skuteczność wystąpień kierownictwa w jednej z agencji reklamowych, w których to wystąpieniach menedżerowie przekonywali, czemu ta agencja jest szczególnie przyjazna dla pracowników, jak ich wyróżnia itp., (1985/91: 89): „Dominacja menedżerów w praktyce podtrzymywana jest nie przez pęd pracowników-słuchaczy (...) prezentacji do wydajnej pracy, ale przez akceptację tak zdefiniowanego terytorium przez zatrudnionych. Kultura, tworzenie i istnienie terytorium w świadomości jest tu mechanizmem kontroli”.

Podobnie twierdzi Trice (1993), pisząc, że legitymacja menedżerów do kierowania pracą specjalistów często uzasadniana jest pseudoracjonalnością, tworzeniem paranaukowych teorii – jak np. „naukowe zarządzanie” Taylora. Wypowiedzi moich rozmówców o karierze i percepcja dużych firm wskazują na fakt, że programiści być może są szczególnie wyczuleni na cha-

rakterystyczne dla kultury menedżerskiej zapędy do „propagandy”, definiowania wspólnej świadomości pracowników. We współczesnej literaturze menedżerskiej tendencja do „zarządzania kulturą” jest bardzo widoczna, dominuje tzw. podejście kliniczne (Schein, 1985), polegające na stosowaniu różnych socjotechnik, by umożliwić menedżerom taki wpływ na kulturę, który wywoła dodatkowy pozytywny efekt ekonomiczny. W literaturze z zakresu zarządzania projektami informatycznymi tego rodzaju podejście można zobrazować następująco (Prager, 1999: 12): „Dla specjalistów z branży informatycznej i nieinformatycznej istnieją osobne kultury organizacyjne, opierające się na różnych zachowaniach organizacyjnych. (...) Te osobne, równoległe światy wpływają na każdego, kto ma do czynienia ze społecznością IT. Menedżerowie muszą zdecydować, czy wspieranie (lub przeciwstawianie się) osobnej kulturze profesjonalnej techników pomoże osiągać cele przedsiębiorstwa”.

Jednak, jak zauważa wielu autorów (por. Kostera, 1996), w organizacjach powszechna jest antynomia między wartościami deklarowanymi (reprezentującymi retorykę menedżerską, nazwaną przez moich rozmówców „propagandą”) a wartościami faktycznie przestrzeganyimi, będącymi w użyciu. Programiści odrzucają menedżerską retorykę – sami na własny sposób definiują swoją rolę w organizacji.

Jednak tak jak u Rosena (1986/91) istotą emancypacji kultury programistów nie jest tylko negowanie retoryki kierowniczej, lecz raczej także tworzenie własnych, alternatywnych kategorii definiujących ich rolę i miejsce w organizacji. To swoiste zderzenie cywilizacji czy też *culture-clash* – i trudno chyba znaleźć inną grupę zawodową, która równie skutecznie definiowałaby kulturę organizacji, wpływając na całą firmę.

5. Wojna artefaktów na przykładzie ubioru

Znamiennym przejawem ignorowania menedżerskiej symboliki i siły kultury programistów może być odrzucenie powszechnego w większości firm *dress-code*, wpajanego pracownikom wzorca poprawnego ubierania się. O ile moi rozmówcy byli powszechnie przekonani, że z zawodem programisty nie wiąże się żaden konkretny styl ubioru, o tyle wszyscy ubrani byli dość luźno (warto zauważyć, że w 2005 roku zostali ochrzczeni w Australii *najgorzej ubraną grupą zawodową* – Hearn, 2005). Co więcej, uważali po-

wszechnie, że w ich firmie nie występują normy dotyczące ubioru, choć większość wyrażała upodobanie dla swobodnego stroju.

Warto jednak zauważyć, że badani programiści jednomyślnie twierdzili, że w sytuacjach, kiedy przychodzą do pracy w garniturze, spotykają się z żartami ze strony kolegów. Drwiny kolegów i docinki z całą pewnością można uznać za przejaw normy kulturowej – często niezwykle okrutnie dokonują one socjalizacji i nadania roli społecznej (Konecki, 1992). Norma przyjmowania artefaktów kojarzonych z dużymi firmami lub z kulturą menedżerską. Co ciekawe, menedżerowie, z którymi rozmawiałem, w pełni akceptują ten standard ubioru (SandM):

[P:] Zauważyłem, że wszyscy informatycy tutaj w firmie, z którymi rozmawiałem, byli w strojach takich luźnych dość.

[O:] Tak. To znaczy w zasadzie chyba, jeżeli ktoś nie wychodzi do klienta, to oby tylko w szortach nie przychodził. Także ja nie mam... ja do tego nie przywiązuję jakiejś specjalnej wagi, że ktoś ma być po prostu w garniturze przychodzić. Ubiera strój, w którym jest mu wygodnie po prostu, schludnie ubrany i tyle, no nie.

[P:] Ale z kolei z Panem jak siedzę i widzę, że Pan jest raczej w stroju takim no... [uwaga od badacza – menedżer ubrany był w garnitur i pod krawatem]

[O:] Znaczący, no generalnie tak, no bo po pierwsze mam oczywiście często spotkania z kimś z zewnątrz czy z klientami, więc wypada być przynajmniej w miarę sensownie ubranym, a po drugie nawet jak nie mam w kalendarzu, to zawsze coś może wypaść, no i się już parę razy naciąłem, więc staram się już pilnować, żeby po prostu być zawsze gotowym krótko mówiąc do wyjścia, co nie. Także no taka jest moja po prostu rola, co nie i generalnie ci koledzy, którzy wychodzą do klienta i im to grozi to od razu Pan pozna. Przejdzie Pan po pokojach to będzie widać, że ten klient, ci koledzy często wychodzą do klienta.

Mój rozmówca był byłym programistą – jak sam mówi na początku „się parę razy naciął”, stosując normę ubioru programisty, a samemu pełniąc już rolę kierowniczą. Co najbardziej interesujące, we wszystkich trzech firmach noszenie garnituru było społecznie akceptowane tylko w ściśle określonych kontekstach zewnętrznych – spotkania z klientem (Sand7):

[P:] A Pan często nosi garnitur?

[O:] Ja sporadycznie i wcześniej mam info, że idę do klienta, tak. Powiedzmy dzień wcześniej, w związku z tym wbijam się w garnitur, ja wtedy wbijam się w garnitur nie luźno, idę do klienta, a na zewnątrz do klienta to trzeba ładnie wyglądać, to nawet nie jest tak, że my

idziemy na spotkanie z panem w garniturze a klient niekoniecznie, ale my musimy wyglądać no na profesjonalistów i przedstawiamy w tym momencie firmę. I to z prostej rzeczy wynika.

O ile zatem moi rozmówcy odrzucali menedżerskie normy ubioru wewnątrz organizacji, o tyle akceptowali je, gdy mieli reprezentować firmę przed klientem. Jak pisze Rothenbuhler (1998/2003: 35): „Skondensowane symbole religijne – a dotyczy to również tych związanych z postawami patriotycznymi czy nacjonalistycznymi – eksplodują znaczeniem, gdy zostają wyzwolone w sytuacji rytualnej. Zwykle przedmioty, jak flagi, uniformy, krzyże i szaty zdają się wypełniać znaczeniem sytuacje, w których się pojawiają. Zalewają świadomość, zmniejszając znaczenie ego, wypełniając ją natomiast doświadczanym nieuchronnym przepływem znaczenia”.

Podobnie można powiedzieć o organizacyjnym *dress-code* – jest on narzędziem umacniania roli organizacyjnej, zacierania cech indywidualnych. Dlatego właśnie wobec organizacji programiści manifestują wolność ubioru, zgadzają się natomiast z niej zrezygnować poza organizacją. Potwierdza to także, jak sądzę, występowanie zderzenia kulturowego i rywalizowania w wielu firmach dwóch systemów wartości i obyczajów – menedżerskiego i programistycznego.

Zmianę tego układ wnosi dopiero klient – w relacjach z nim programiści skłonni są akceptować pewną standaryzację ubioru, choć narzuconą już nie tyle przez samą organizację, ale także przez domniemane normy pozaorganizacyjne (oczekiwania klienta). Dlatego trójkąt menedżer–programista–klient może stanowić ciekawą oś dla dalszej analizy wyników badań (podobny problem opisuje Postuła, 2007).

6. Podsumowanie

Zdaniem Halla (1986), opozycja organizacja (władza menedżerów, wynikająca z formalnego mandatu właścicieli)–profesja (wiedza ekspertów) występuje w większości firm. Potwierdza to Trice (1993), pisząc, że choć każda z kultur profesjonalnych jest unikalna i ma inną charakterystykę, większość z nich ma jedną wspólną cechę – są w opozycji do kultury menedżerów. Podstawowym założeniem jest, że „[t]o członkowie danych profesji, a nie menedżerowie, powinni kontrolować najważniejsze aspekty pracy. Stoi to w jaskrawej sprzeczności z założeniem menedżerów – że to kierownicy powinni posiadać większość prerogatyw” (Trice, 1993: 41).

Np. wśród księgowych funkcjonuje powiedzenie (Pondy w: red. Pondy, Frost i in., 1983), że „zawodem księgowego jest chronienie firmy przed menedżerami”. Opozycja profesja–menedżer staje się coraz bardziej widoczna (warto tu choćby odnotować oszałamiający sukces komiksu „Dilbert”, opisyującego perypetie niekompetentnych menedżerów i profesjonalnych inżynierów), zwłaszcza wobec upowszechnienia standardów edukacji menedżerskiej i ich konsolidacji kulturowej, opisywanej np. przez Koźmińskiego (1999).

Kultura zawodowa programistów definiowana jest w opozycji do menedżerów, są oni awangardą „nowego świata pracy ery elektronicznej” (Gephart, 2002). Ich sytuację w odniesieniu do kierownictwa trafnie oddaje Zabusky (1997: 129): „To zestawienie nie jest prostym porównaniem dwóch komplementarnych form. Przeciwnie, zawiera w sobie walkę o legitymizację, władzę i autonomię, wewnątrz współczesnych organizacji”.

W przeciwieństwie do tradycyjnej hierarchii, programiści swoją pozycję budują w oparciu o autorytet wypływający z wiedzy – w odróżnieniu od wielu innych grup pracowników wiedzy (*knowledge workers*) ich kompetencje są dla menedżerów całkowicie nieprzejryste, mają charakter dyskretny i ukryty (*tacit*). To powoduje, że stanowią niezwykle interesującą i unikatową grupę zawodową (Kociatkiewicz i Kostera, 2003; Czarkowska, 2006).

Funkcjonują w przestrzeni wiedzy praktycznej, czyli „Trybu 2” w rozumieniu Nowotnego, Scott and Gibbonsa (2003), ponieważ aplikacja systemów informatycznych ma zawsze charakter mocno kontekstowy. Skutkuje to między innymi głębokim kryzysem zaufania w projektach IT (Latusek, 2007), „wojną o terminarze” (Jemielniak, 2005), kryzysem tożsamości (Alvesson, 1993) i in. Opisany konflikt między grupami zawodowymi ma zatem głębokie implikacje, które powinny stać się przedmiotem dalszych badań.

BIBLIOGRAFIA

Alvesson, M. (1993) „Organizations As Rhetoric: Knowledge-Intensive Firms And The Struggle With Ambiguity”, *Journal of Management Studies*, Tom 30, Nr 6.

- Barley, S. i Kunda, G. (2004) *Gurus, Hired Guns, and Warm Bodies: Itinerant Experts in a Knowledge Economy*, Princeton: Princeton University Press.
- Boje, D. M. (1991) „The Storytelling Organization: A Study of Story Performance in an Office-Supply Firm”, *Administrative Science Quarterly*, 36 (1), str. 106–126.
- Czarkowska, L. D. (2006) „Nowa profesja: informatycy”, w: *Problemy zarządzania*, nr 1 (11), str. 178–194.
- Gephart, R. P. (2002) „Introduction to the brave new workplace: organizational behaviour in the electronic age”, *Journal of Organizational Behavior*, tom 23, str. 327–344.
- Greenwood, D. i Levin, M. (1998) *An Introduction to Action Research*, Beverly Hills-London-Delhi: SAGE Publications.
- Hall, R. H. (1986) *Dimensions of Work*, Beverly Hills-London-Delhi: SAGE Publications.
- Hearn, L. (2005) „IT workers dubbed ‘worst dressed’”, *The Sydney Morning Herald*, 17 Nov. 2005, <http://www.smh.com.au/articles/2005/11/17/1132016909640.html>
- James, G. (1986) *The Tao of Programming*, Infobooks, Santa Monica.
- Jemieliński, D. (2002) „Kultura – odkrywana czy konstruowana?”, *Master of Business Administration*, no 2 (55), str. 28–30.
- Jemieliński, D. (2005) „Kultura – zawody i profesje”, *Prace i Materiały Instytutu Studiów Międzynarodowych SGH* nr 32, str. 7–22.
- Kidder, T. (1981) *The Soul of A New Machine*, New York: Avon Books.
- Kociatkiewicz, J. i Kostera, M. (2003) *Shadows of Silence* w: „Ephemera” nr 4/3, str. 305–313.
- Konecki, K. (1992) *Nowi pracownicy a kultura organizacyjna – studium folkloru fabrycznego*, Łódź: Wydawnictwo Uniwersytetu Łódzkiego.
- Konecki, K. (2000) *Studia z metodologii badań jakościowych*, Warszawa: PWN.
- Kostera, M. (1996) *Postmodernizm w zarządzaniu*, Warszawa: PWE.
- Kostera, M. (2003) *Antropologia organizacji. Metodologia badań terenowych*, Warszawa: PWN.
- Koźmiński, A. K. (1999) *Kultura menedżerska* w: *Humanistyka przełomu wieków*, pod red. J. Koźmieleckiego, Warszawa: Wydawnictwo Akademickie „Żak”.
- Koźmiński, A. K. (2004) *Zarządzanie w warunkach niepewności*, Warszawa: PWN.
- Kunda, G. (1992) *Engineering Culture: Control and Commitment in a High-Tech Corporation*, Philadelphia: Temple University Press.
- Kunda, G. i Van Maanen, J. (1999) „Changing Scripts at Work: Managers and Professionals” *Annals of the American Academy of Political & Social Science*, Jan, tom 561, str. 64–81.

- Latour, B. (1986) „The powers of association” in: J. Law (ed.) *Power, Action and Belief – A New Sociology of Knowledge?*, London, Boston, Henley: Routledge&Kegan Paul.
- Latuszek, D. (2007) When trust does not matter: The study of communication practices between high-tech companies and their clients in the environment of distrust, w: Brennan, L. and Johnson, V. *Computer-Mediated Relationships and Trust: Managerial and Organizational Effects* Hershey, PA: Idea Group Publishing.
- Nowotny, H., P. i Gibbons, M. (2003) „Mode 2’ revisited: The New Production of Knowledge”, *Minerva* nr 41, str. 179–194.
- Pondy, L. R., Frost, P. J., Morgan, G. i Dandridge, T. C. (ed.) (1983) *Organizational symbolism*, Greenwich-Connecticut: JAI Press.
- Prager, K. P. (1999) „Organizational Culture and the IT Professional”, *Information Systems Management*, tom 16 nr 2, str. 12–18.
- Postuła, A. (2007) *Rola informatyków w organizacjach*, niepublikowana praca doktorska obroniona na Wydziale Zarządzania Uniwersytetu Warszawskiego.
- Rosen, M. (1985/91) „Breakfast at Spiro’s: Dramaturgy and Dominance”, Frost, P. J. Moore, L. F et. al. *Reframing Organizational Culture*, Newbury Park–London – New Delhi: SAGE.
- Rothenbuhler, E. W. (1998/2003) *Komunikacja rytualna. Od rozmowy codziennej do ceremonii medialnej (Ritual Communication. From Everyday Conversation to Mediated Ceremony)*, Kraków: Wydawnictwo Uniwersytetu Jagiellońskiego.
- Schein, E. (1985) *Organizational culture and leadership. A dynamic view*. San Francisco, CA: Jossey-Bass.
- Schein, E. (1991) *The Role of the Founder in the Creation of Organizational Culture* w: Frost, P. J, Moore, L. F et. al. ‘Reframing Organizational Culture’, Newbury Park – London – New Delhi: SAGE
- Sievers, B. (1990) „Zombies or people – What Is the Product of Work?”, in: Turner, B. A. (ed.) *Organizational Symbolism*, Berlin–New York: De Gruyter
- Trice, H. M. (1993) *Occupational Subcultures in the Workplace*, Ithaca, New York: ILR Press
- Trice, H. M. i Beyer, J. M. (1993) *The Cultures of Work Organizations*, Englewood Cliffs, New Jersey: Prentice Hall
- Ullman, E. (1996) „Out of Time: Reflections on the Programming Life”, *Educom Review* tom 31, nr 4, Jun/Aug 1996
- Weick, K. E. (1969/79) *The Social Psychology of Organizing*, Reading, Massachusetts: Addison-Wesley

Zabusky, S. E. (1997) „Computers, clients, and expertise: negotiating technical identities in a nontechnical world” in: Barley, S. R. and Orr, J. E. *Between Craft and Science*, Ithaca–London: Cornell University Press

Dariusz Jemielniak

PERCEPTION OF MANAGERS IN THE EYES OF SOFTWARE ENGINEERS

S u m m a r y

The article discusses the perception of managers in the eyes of software engineers, by the use of professionalization and occupational culture theory. The programmers' perception of career, organizations and management is depicted, basing on unstructured open-ended interviews. The two vocations are confronted and their culture clash, as described by the interviewed software engineers, analyzed. The study is qualitative, performative and interpretive.

key words: occupational culture, software engineers, organizational ethnography, organizational conflict, workplace stereotypes