

Adaptacja nowego węzła do sieci śledzenia stosowanej w modelu rozwijania oprogramowania opartym na teorii złożoności systemów



Autor: *Piotr Piotrowski*

05

O Autorze:

Pracownik wiedzy, Inżynier, Badacz, Działacz.

Inżynier Testów w dziale Industrial R&D przedsiębiorstwa Tieto Poland, gdzie zajmuje się m. in. czynnościami testowymi w przemyśle motoryzacyjnym.

Posiada doświadczenie głównie w dziedzinie: badań i rozwoju, telekomunikacji, informatyki, jakości i samoprzywództwa. Zaproponował dwa patenty. Jest członkiem i/lub wspiera około trzech organizacji non-profit.

Uczestniczył w rozmaitych projektach i przedsięwzięciach, głównie międzynarodowych, w tym naukowych.



Posiada zaawansowany stopień rozwoju osobistego. Uzyskał bardzo wysokie miejsce w krajowym programie edukacyjnym dla kadry menedżerskiej i wygrał jedną z edycji krajowego konkursu o prestiżową praktykę. Zdobył łącznie około czterdziestu certyfikatów z: zarządzania, angielskiego oraz testowania oprogramowania.

Posiada zaawansowany stopień rozwoju osobistego. Uzyskał bardzo wysokie miejsce w krajowym programie edukacyjnym dla kadry menedżerskiej i wygrał jedną z edycji krajowego konkursu o prestiżową praktykę. Zdobył łącznie około czterdziestu certyfikatów z: zarządzania, angielskiego oraz testowania oprogramowania.

Jest autorem ponad 25 publikacji i 1 monografii. Posiada wykształcenie w: zarządzaniu (niezależne opanowanie programu doktoratu z administracji biznesu), technologii (zaliczone trzy lata doktoratu z telekomunikacji).

Zainteresowania to: rozwój osobisty, działalność naukowa i wiele innych.

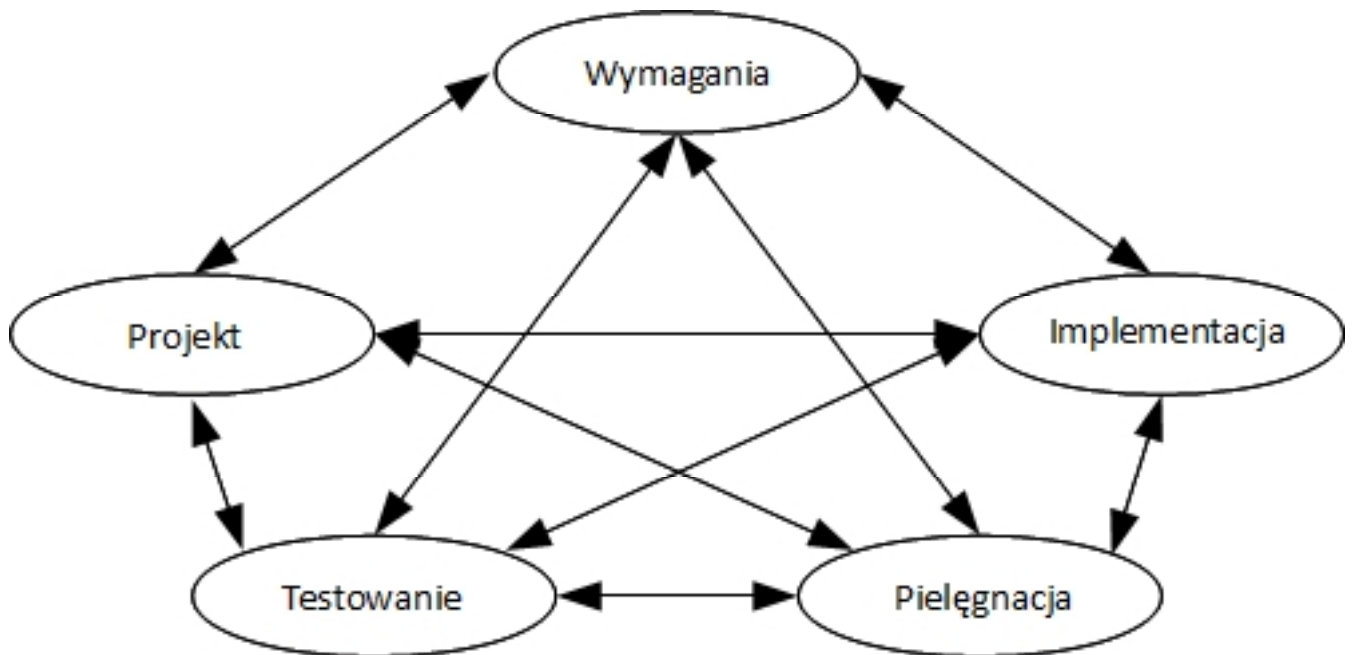
Singiel, dwie siostry.

Streszczenie

Artykuł zawiera propozycje rozwiązania problemu badawczego w postaci: „zapewnienie optymalnej współpracy nowego węzła z siecią śledzenia stosowaną w rozwijaniu oprogramowania według modelu opartego na teorii złożoności systemów”. W pracy wyjaśniono terminy: modelu rozwijania oprogramowania opartego na teorii złożoności systemów jako generacji modeli opracowanych później niż rozwiązania: sekwencyjne, iteracyjne i przyrostowe, sieci śledzenia będącej uogólnieniem macierzy śledzenia, a następnie przedstawiono rozwiązania postawionego problemu w warstwie zadaniowej i społecznej.

Złożony model rozwijania oprogramowania

W [NSE11] wyróżnia się następujące główne zasady złożoności systemów: nieliniowość, holizm, dynamizm, otwartość, samoorganizacja, samoadaptacja.



Rysunek 1 Ogólny model złożonego systemu rozwijania oprogramowania – opracowanie własne.

W modelu rozwijania oprogramowania opartym na teorii złożoności systemów każda faza rozwijania oprogramowania, będąca komponentem systemu, łączy się potencjalnie z co najmniej jedną, inną fazą, co pokazano na Rys. 1.

odpowiadają ludzie o określonych umiejętnościach, wyposażeni w zasoby do realizacji stawianych przed nimi zadań, którzy współpracują za sobą i tworzą sieć społeczną, czyli węzły mają postać (ludzie, umiejętności, zasoby, zadania).

się z: z innym testerem, deweloperem, kierownikiem projektu. Jednak należy być otwartym na możliwość relacji także z pozostałymi specjalistami rozwijania oprogramowania – faza testowania może mieć związki z innymi fazami w koncepcji modelu rozwijania oprogramowania opartego na teorii złożoności systemów, a ilość i jakość tych relacji zależy od wielu czynników, np. relacje z kierownikiem projektu mogą być częstsze wskutek nieobecności w projekcie kierownika testów, który dzieli się obowiązkami zarządczymi z tym pierwszym.

Sieci śledzenia

P W modelu rozwijania oprogramowania opartym na teorii złożoności systemów, śledzenie (ang. traceability) jest dwukierunkowe i odbywa się potencjalnie pomiędzy wszystkimi fazami rozwijania. Tego typu śledzenie ma postać wielowymiarowej sieci śledzenia (ang. traceability network), która bierze pod uwagę więcej związków pomiędzy komponentami systemu niż tradycyjna macierz śledzenia (ang. traceability matrix). W dwuwymiarowej macierzy śledzenia – dla przypomnienia – kojarzy się wymagania z produktami pracy, które mają spełniać te wymagania, czyli dotyczy ona związków z fazą wymagań.

W dalszej części artykułu zostanie dokładniej rozpatrzona faza testowania ze względu na największe doświadczenia autora w tym zakresie i zgodność z tematyką konferencji. Założono, że za fazę testowania odpowiada tester oprogramowania, który realizując swoje zadania wchodzi w interakcję z osobami odpowiadającymi za inne fazy rozwijania oprogramowania. Inaczej rzecz przedstawiając, rozwijanie oprogramowania następuje według fragmentu modelu, przedstawionego na Rys. 1, i w związku z tym tester oprogramowania może potencjalnie wchodzić w interakcję ze wszystkimi innymi specjalistami od rozwijania oprogramowania: faza wymagań – analityk oprogramowania, faza projektu – architekt (lub projektant) oprogramowania, faza implementacji – deweloper oprogramowania, faza pielęgnacji – różne z tych osób w zależności od potrzeb. Poza tym w projekcie występuje też kierownik projektu.

Relacje (inaczej połączenia) między węzłami sieci można rozpatrywać w przenikających się płaszczyznach – zadaniowej i społecznej, a także w wielu kierunkach (Rys. 1).

Adaptacja nowych węzłów do sieci śledzenia

Problem adaptacji nowego węzła do sieci śledzenia w kompleksowy sposób można rozpatrywać jako: jego przygotowanie się do połączenia z istniejącą siecią oraz interakcję z nią, która pociąga za sobą zmiany w węźle oraz całej sieci.

Przed dołączeniem do sieci rozwijania oprogramowania, tester przechodzi szkolenie. Następnie po włączeniu w sieć, i w trakcie interakcji z innymi węzłami, oczywiście w pewnym stopniu dalej się uczy i rozwija, w tym w trakcie wykonywania swoich zadań. Tester przy realizacji swoich obowiązków najczęściej komunikuje

Adaptacja nowych testerów w warstwie zadaniowej

Nowego testera szkoli się w celu efektywnego i wydajnego wykonywania przez niego przyszłych zadań. Proces szkoleniowy zawiera [Veen10], [Bath11]: określenie potrzeb szkoleniowych, sporządzenie planu szkolenia, przygotowanie materiałów szkoleniowych, dostarczenie szkolenia, zapis przebiegu szkolenia, oszacowanie efektywności szkolenia. Tester musi być wyposażony w różne narzędzia swojej pracy i być przeszkolony z metod ich stosowania.

Za każdą z faz rozwijania oprogramowania w projekcie, widocznych na Rys. 1,

Testerom przydzielane są odpowiednio szczegółowe, określone w czasie, ambit-

ne, realne i mierzalne cele [Bath11], aby doprowadzić do wzrostu ich umiejętności w ramach planu rozwoju, w tym pokrycia ewentualnych luk w kompetencjach. Cele powinny być wzajemnie uzgodnione pomiędzy kierownikiem a testerem. Są one ustalane okresowo lub na bieżąco, a następnie monitorowane i oceniane. Różne cele mogą być ustalane dla poszczególnych testerów – podejście indywidualne lub dane cele mogą być wspólne dla grupy testerów lub dla innej, mieszanej konfiguracji pracowników – podejście zespołowe. Zespoły mieszane, zbudowane z różnego rodzaju specjalistów do spraw rozwijania oprogramowania, stwarzają testerowi dalsze możliwości szkoleniowe.

W miarę rozwoju testera, zwłaszcza po pewnym czasie od momentu jego podłączenia do sieci rozwoju oprogramowania, przydziela mu się coraz bardziej złożone, zaawansowane cele. Od chwili włączenia do projektu rozwijania oprogramowania zmienia się też charakter szkoleń, wzrasta ilość wiedzy nabywanej wskutek wykonywania obowiązków w projekcie, a maleje z reguły udział wiedzy przyswajanej na szkoleniach o charakterze teoretycznym – pokazowym.

Testerzy mogą się specjalizować pełniąc określone role, z którymi wiąże się dany zakres odpowiedzialności [Bath11]. Przykładowo analityk testów (ang. test analyst) projektuje testy stosując techniki czarnej skrzynki (ang. black box), jak choćby podział na podzbiory równoważności (ang. equivalence partitioning). W celu zaprojektowania przypadków testowych na potrzeby testów systemowych, analityk testów potrzebuje między innymi: wymagań określających zakładaną pracę systemu, szablonu przypadku testowego, znajomości narzędzi testowych, w tym narzędzia do zarządzania testami, w którym umieści przypadki testowe oraz oczywiście wiedzy o technologii/produkcji, dla którego projektuje przypadki testowe, a także zakresu i terminu wykonania tejże pracy. Innym częstym zadaniem analityka testów jest wykonanie zaprojektowanych przypadków testowych oraz zaraportowanie wyników tego wykonania, w tym również zgłoszenie ewentualnych błędów w pracy testowanego produktu.

Bardziej specyficznie, analityk testów może również pełnić rolę konsultanta w pracach nad tworzeniem wymagań systemu, czy angażować się w aktywności



związane z rozwijaniem oprogramowania na niższym poziomie – to znaczy na podstawie swego doświadczenia wypowiedzieć się oraz wspomagać także fazy projektowania, czy implementacji. Analityk testów może mieć również związki z fazą pielęgnacji przy testowaniu wyników naprawy błędów w produkcie, czy z kierownictwem projektu przy określaniu planu, a nawet strategii testów. W zakresie, zwłaszcza swojego szkolenia, w obszarach związanych z zagadnieniami testowania, analityk testów kontaktuje się również z innymi, posiadającymi większą wiedzę dotyczącą danego zagadnienia, testerami. Powyższe przykłady zadań, w jakich może brać udział analityk testów, będący reprezentantem fazy testowania pokazują, że faza ta ma potencjalnie związki także z innymi fazami rozwijania oprogramowania, a przedstawiciele poszczególnych faz tworzą sieć społeczną projektu, choć oczywiście relacje między nimi są różne, chociażby pod względem częstotliwości kontaktów. W uproszczeniu można założyć, że jeśli chodzi o zadania analityka testów na poziomie testów systemowych, to najczęściej potrzebuje on kontaktów z: innymi testerami, deweloperami oprogramowania oraz kierownictwem projektu. W kolejnym podpunkcie artykułu zostaną przeanalizowane właśnie takie relacje.

Udana adaptacja analityka testów, testera, jak i każdego innego pracownika, jeśli chodzi o warstwę zadaniową środowi-

ska pracy, powinna odznaczać się tym, że zatrudniony przejawia wiele z cech w postaci:

- „Pracownik przed realizacją zadania możliwie najbardziej szczegółowo zapoznaje się z oczekiwaniami przełożonego w zakresie końcowych efektów działania
- Pracownik samodzielnie, lub pytając kompetentną osobę, potrafi uzupełnić brakującą wiedzę niezbędną do realizacji zadań, jest otwarty na podwyższenie własnych kompetencji
- Pracownik z zaangażowaniem i odpowiedzialnością realizuje powierzone mu zadania
- Pracownik ocenia swoją pracę przez pryzmat osiągniętych efektów, a nie włożonego wkładu pracy
- Pracownik nie narzeka na pojawiające się problemy, tylko szuka ich rozwiązania
- Pracownik realizując powierzone zadania zgodnie z zalecanymi procedurami, jest jednocześnie aktywny w poszukiwaniu rozwiązań służących ulepszeniu metod i efektów pracy
- Pracownik prezentuje pozytywne nastawienie do realizowanych zadań
- W przypadku, gdy pracownik popełni błąd potrafi się do niego przyznać.” [Lista stanowi cytowanie oczekiwań pracodawców wymienionych w rozdz. 5 [Prze11].]

Adaptacja nowych testerów w warstwie społecznej

Każdy członek organizacji, w tym nowy tester posiada określony typ osobowości oraz preferuje pewne role do pełnienia w przyszłej sieci projektu rozwijania oprogramowania. Znajomość tych elementów pozwala na [Bath11]: zrozumienie własnych zachowań, preferowanych stylów uczenia oraz postępowania z problemami i sytuacjami stresowymi. Kierownictwo może wykorzystać je, obok kompetencji twardych (jak np. wiedza o danej domenie testowej, czy technologii), podczas zarządzania siecią specjalistów do spraw rozwoju oprogramowania, choćby podczas budowania i rozwijania zespołu. Typ osobowości może być określony za pomocą wskaźnika Myers-Briggs Type Indicator, MBTI. W MBTI każdy z 16-tu typów jest określony za pomocą 4 liter, gdzie każda z liter może przyjąć jedną z dwóch przeciwstawnych wartości. Na podstawie Tab. 1. [CaAh10], [Capr11] widać, że na przykład w przypadku testerów oprogramowania, w związku ze specyfiką realizowanych zadań, najczęściej pojawiają się litery S (Poznanie) i J (Osądzanie) w typach osobowości; w sytuacji deweloperów – I (Introwersja), S, T (Myślenie); a jeśli chodzi o kierowników projektów, według [MPM11] potencjalnie największy sukces w kierowaniu projektem technicznym zapewnia osobowość określona typem ESTJ, natomiast na podstawie [Schw06] widać, że w typach osobowości najbardziej dopasowanych do przewodzenia projektowi występują najczęściej litery T i J.

Jeżeli określone grupy zawodowe charakteryzują się pewnymi typami osobowości i dany typ osobowości komunikuje się w dany sposób, to jest możliwe powiązanie stylu komunikacji z różnymi rodzajami stanowisk. Uświadomienie sobie przez testera istnienia preferowanych stylów komunikacji interpersonalnej poszczególnych przedstawicieli specjalistów do rozwijania oprogramowania i kierownictwo projektu powinno ułatwić mu zaadaptowanie się w warstwie społecznej, ponieważ komunikacja interpersonalna jest ważną umiejętnością mięką warunkującą pożądane funkcjonowanie w sieci społecznej projektu rozwijania oprogramowania. Szerzej, znajomość różnic i podobieństw typu osobowości między osobami tworzącymi daną relację pozwala przewidzieć i radzić sobie z przyszłymi trudnościami w relacjach. Generalnie, jeśli chodzi o relacje między współpra-

cownikami, to są one lepsze gdy typy osobowości i przykładowo preferowane style komunikacji interpersonalnej – jako ich pochodna – są najbardziej zbliżone.

W komunikacji między testerami ryzyko sytuacji problemowych jest więc najmniejsze przy powyższym założeniu, że tą podsić społeczną tworzą osoby, dla których już dwie spośród czterech płaszczyzn wskaźnika MBTI są identyczne. Z kolei w interakcji między testerami a deweloperami, czy kierownikami projektu występuje natomiast jedna wspólna litera kodu MBTI. Jednak dane wyrażone między innymi za pomocą Tab. 1 stanowią jeden z dosyć prawdopodobnych scenariuszy, a do pełnego zrozumienia i częściowego dopasowania stylów komunikacji interpersonalnej, należałoby podać charakterystykę tychże stylów dla wszystkich typów osobowości według MBTI. Jest ona opisana przykładowo w [FILE05] w postaci:

- „Ekstrawersja (E) – osobiste zebranie się w celu werbalnego zaprezentowania przemyśleń,
- Introwersja (I) – pomoc osobie narysować problem, a następnie dać jej nieco czasu na prywatne odniesienie się do wiadomości,
- Poznanie (S) – zaprezentowanie namacalnych faktów, przykładów, danych i rzeczywistych doświadczeń w celu przedstawienia własnego punktu widzenia,
- Intuicja (N) – zaoferowanie przeglądu „wielkiego obrazu”, zaprezentowanie koncepcji, które są kluczowe dla dyskusji,

- Myślenie (T) – zaprezentowanie argumentów, które wynikają z racjonalnej analizy faktów, odnoszą się do rozumu,
- Odczuwanie (F) – mówienie od serca, odnoszenie się do sformułowań odnoszących się do wartości i decyzji podejmowanych na podstawie przeczucia
- Osądzanie (J) – prezentowanie informacji w sposób uporządkowany, dyskusja w kierunku rozstrzygnięcia i zamknięcia,
- Obserwacja (P) – zezwolenie na otwartą dyskusję, elastyczne podejście do porządku dyskusji.” [Cytat jest tłumaczeniem zawartości tabeli 3-1 zawartej na str. 45 w [FILE05].]

Na komunikację interpersonalną wpływają również wartości, w tym wartości związane z pracą (ang. work values), które na podstawie danych pochodzących z programu O*NET (ang. Occupational Information Network) [ONO12] zestawiono w Tab. 2.

Definicje wymienionych powyżej wartości są następujące [ONO12]:

- Osiągnięcie (ang. Achievement) – charakteryzuje zawody zorientowane na wyniki, w których pracownicy mogą stosować swoje najsilniejsze strony oraz daje się im odczuć smak dokonań. Odpowiadającymi potrzebami są: zdolność, spożytkowanie i osiągnięcie;
- Warunki pracy (ang. Work Conditions) – obejmuje zawody oferujące bezpieczeństwo i dobre warunki pracy. Związane z tym potrzeby to: aktywność, wynagrodzenie, niezależność, bezpieczeństwo, różnorodność i warunki pracy;

MBTI Dimensions	Software Life Cycle Phases				
	Systems Analysis	Software Design	Programming	Testing	Maintenance
Extroversion (E)	√				
Introversion (I)			√		
Sensing (S)			√	√	√
Intuition (N)		√			
Thinking (T)		√	√		
Feeling (F)	√				
Judging (J)				√	
Perceiving (P)					√

Tabela 1. Dominujące litery kodu MBTI dla poszczególnych grup specjalistów do spraw rozwijania oprogramowania na podstawie [CaAh10], [Capr11].

	Zawód		
	Tester	Deweloper	Kierownik systemów informacyjnych (IT)*
Wartości pracy:	1. Osiągnięcie 2. Warunki pracy 3. Niezależność	1. Osiągnięcie 2. Warunki pracy 3. Uznanie	1. Warunki pracy 2. Wsparcie 3. Osiągnięcie

* Kierownik systemów informacyjnych (IT) jako przybliżenie kierownika projektu

Tabela 2. Dominujące wartości w środowisku pracy dla poszczególnych grup specjalistów do spraw rozwijania oprogramowania i kierownictwa na podstawie [ONO12].

- **Niezależność** (ang. Independence) – zawody pozwalające pracownikom na pracę według własnego zamysłu i podejmowanie decyzji. Potrzeby: kreatywność, odpowiedzialność i autonomia.
- **Uznanie** (ang. Recognition) – zawody oferujące awans, możliwość przewodzenia, odbierane jako prestiżowe. Potrzeby to: awans, władza, uznanie, status społeczny.
- **Wsparcie** (ang. Support) – zawody związane ze wsparciem pracowników przez kierownictwo. Potrzeby: polityki przedsiębiorstwa, nadzór w zakresie zasobów ludzkich oraz technologii.

Znajomość, związanych z potrzebami, wartości pracy ułatwia zrozumienie w procesie aktywnego słuchania odnośnie motywów działania przedstawicieli poszczególnych zawodów. Zbieżność wartości pracy [PaMc06] między obydwoma węzłami sieci warunkuje wysoki poziom satysfakcji ze wzajemnych relacji. Jednak należy pamiętać, że teoria wartości pracy kładzie nacisk przede wszystkim na indywidualne doświadczanie środowiska pracy. Przykładowe zawody, Tab. 2, na liniach tester – deweloper, tester – kierownik IT, różnią się jedną wartością, lecz ważna jest również kolejność wartości. Poszczególne wartości nie wykluczają się wzajemnie w pełni. Specjaliści do spraw rozwijania oprogramowania w relacjach z kierownikami projektów mogą przykładowo doceniać warunki pracy, nie

odrzucać wsparcia, czy umożliwiać realizację celów.

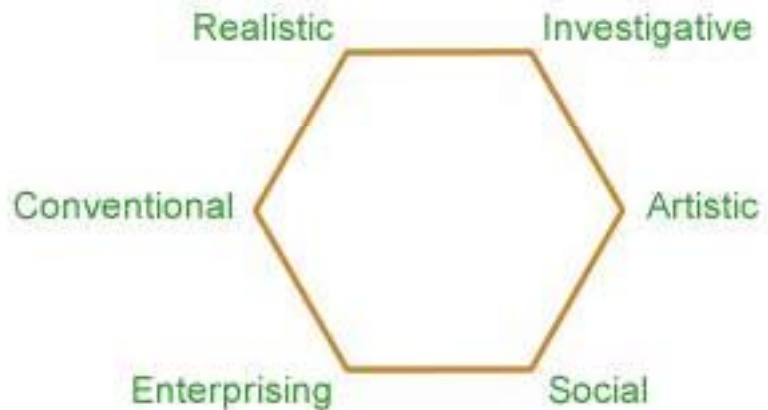
Innym wymiarem charakterystyki zawodów są, skorelowane ze sobą, zainteresowania (ang. Interests), również stosowane w programie O*NET, uwidocznione w Tab. 3.

Wyjaśnienia znaczeń rodzajów zainteresowań wymienionych w Tab. 3 na podstawie [ONO12]:

- **Badawcze** (ang. Investigative) – tego typu zawody polegają na pracy z ideami, wymagając dużej ilości myślenia, wiążą się z poszukiwaniem faktów i zrozumieniem problemów;
- **Konwencjonalne** (ang. Conventional) – zawody wymagające często stosowania procedur, mogą obejmować pra-

cę z danymi i szczegółami w większym stopniu niż z ideami, zwykle występuje ściśle określenie władzy, której polecenia się wykonuje;

- **Realistyczne** (ang. Realistic) – zawody często obejmujące czynności zawierające praktyczne problemy i rozwiązania. Pracownicy posługują się tutaj rzeczywistymi materiałami, takimi jak np. drewno, narzędzia i maszyny. Jest to czasami praca w terenie bez dużej ilości czynności wykonywanych na papierze;
- **Przedsiębiorcze** (ang. Enterprising) – zawody często polegające na uruchamianiu i dbaniu o projekty. Mogą pociągać za sobą przewodzenie ludziom i podejmowanie wielu decyzji. Czasami wymagają podejmowania ryzyka i prowadzenia interesów.



Rysunek 2. Sześciokąt Hollanda. [Capr11].

	Zawód		
	Tester	Deweloper	Kierownik systemów informacyjnych (IT)*
Zainteresowania:	1. Badawcze 2. Konwencjonalne 3. Realistyczne	1. Badawcze 2. Konwencjonalne 3. Realistyczne	1. Przedsiębiorcze 2. Konwencjonalne 3. Badawcze

Tabela 3. Dominujące wartości w środowisku pracy dla poszczególnych grup specjalistów do spraw rozwijania oprogramowania i kierownictwa na podstawie [ONO12].

Zależności między poszczególnymi zainteresowaniami są zobrazowane za pomocą sześciokąta Hollanda – Rys. 2 – według którego zainteresowania przylegające (na przykład Realistyczne – Badawcze) są najbardziej podobne do siebie, natomiast naprzemienne (na przykład: Konwencjonalne – Badawcze) będą miały pośrednie związki. Najdogodniejsza sytuacja występuje, gdy podstawowe zainteresowania są jak najsilniej skorelowane z dalszymi (na przykład: Badawcze – Realistyczne – Konwencjonalne) ze względu na poziom satysfakcji osoby wykonującej dany zawód.

Tab. 3 pokazuje, że nie ma różnic między testerami a deweloperami jeśli chodzi o rodzaj i kolejność zainteresowań, co wskazuje na niskie ryzyko problemów w relacjach między nimi. Kierownicy wykazują natomiast dodatkowo zainteresowania Przedsiębiorcze, choć jako czwartą wartość przyjmuje się dla nich zainteresowania Realistyczne. Zainteresowania wpływają na przykład na tematykę komunikacji. Przykładowo tester, czy deweloper, w relacji z kierownikiem IT, powinien zwracać uwagę na: przestrzeganie harmonogramów, zyskowość, poprawne relacje ze współpracownikami, czy ryzyko działań.

Każdy członek zespołu, określanego w tej pracy jako sieć projektu rozwijania oprogramowania, posiada określone preferencje związane z pracą w grupie, które wpływają na jego zachowania oraz relacje międzyludzkie i są nazywane rolami w zespole (ang. team roles). Jeden z profili ról zespołowych zaproponował Belbin, co przedstawia Tab. 4 [BEL12].

U różnych ludzi występują predyspozycje do pełnienia co najmniej jednej roli. Na podstawie doświadczeń [Vint08] z: testerami, deweloperami i kierownikami projektów, które zestawiono w Tab. 5, widać między innymi, że wszystkie te zawody jako główną przyjmują preferencje Realizatora, a różnice pomiędzy różnego rodzaju specjalistami do spraw rozwijania oprogramowania występują w zakresie dalszych preferencji. Nie-

Rola	Mocne strony	Słabe strony
Filar/Roślina (ang. Plant)	kreatywny, obdarzony wyobraźnią, otwarcie myślący. Generuje idee i rozwiązuje trudne problemy	ignoruje przypadkowość, zbyt pochłonięty aby komunikować się efektywnie
Zaradna dusza (ang. Resource Investigator)	towarzyski, entuzjastyczny, komunikatywny. Eksploruje szanse i rozwija kontakty	zbyt optymistyczny, traci zainteresowanie, gdy przemienie początkowy entuzjazm
Koordinator (ang. Coordinator)	dojrzały, pewny, identyfikuje talent. Wyjaśnia cele, efektywnie deleguje	może być odbierany jako manipulujący, przerzuca swoją część pracy
Animator (ang. Shaper)	lubi wyzwania, pracę w napięciu. Posiada ambicje do pokonywania przeszkód	podatny na prowokację, uraża ludzkie uczucia
Obserwator (ang. Monitor Evaluator)	stateczny, strategiczny, bystry. Widzi wszystkie opcje i dokładnie wyrokuje	wykazuje niedostatki w inspirowaniu innych, może być zbyt krytyczny
Pracownik zespołowy (ang. Team Worker)	współpracujący, spostrzegawczy i dyplomatyczny. Słucha i odwraca spięcia	unika ważnych decyzji oraz konfrontacji
Realizator (ang. Implementer)	praktyczny, rzetelny i skuteczny. Zamienia idee w działania i organizuje pracę	częściowo sztywny, wolno odpowiada na nowe możliwości
Finalizator (ang. Completer Finisher)	skrupulatny, sumienny, pełen troski. Wyszukuje błędy, błyskotliwy i perfekcyjny	skłonny do zbytniego zamartwiania, niechętny do delegowania
Specjalista (ang. Specialist)	wytrwały, automatycznie rozpoczynający, oddany. Dostarcza wiedzę i umiejętności	wnosi wkład w wąskim zakresie, zajmuje się fachowymi szczegółami

Tabela 4. Profile ról zespołowych według Belbina [BEL12].

stety badania obserwacje przeprowadzone przez [Vint08] nie uwzględniają roli Specjalisty, która prawdopodobnie wśród specjalistów do spraw rozwijania oprogramowania może się znaleźć na drugich miejscach, natomiast dla kierowników projektów na dalszych.

W dalszych akapitach zostanie przedstawiony opis relacji dla ról (ściśle według [Belb03]) wymienionych w Tab. 5.

Jeżeli zarówno podwładny (Tester, Deweloper) jak i przełożony (Kierownik projektu) są Realizatorami, to ich wza-

	Testerzy	Deweloperzy	Kierownicy projektów
Rola w zespole	Realizator	Realizator	Realizator
	Pracownik zespołowy	Pracownik zespołowy	Animator
	Finalizator	Filar	Zaradna dusza

Tabela 5. Profile ról zespołowych dla wybranych zawodów według [Vint08].

jemne relacje mogą być gorsze z uwagi na przesyt biurokracji. Z drugiej strony, Realizatorom najlepiej współpracuje się z: Zaradnymi duszami, Finalizatorami oraz Specjalistami. Realizatorzy mogą prowadzić burzliwe dyskusje z innymi Realizatorami oraz Filarami. Realizatorzy preferują sformalizowane stosunki z posłusznymi podwładnymi. Na tym polu najwięcej problemów sprawiają Finalizatorzy z uwagi na ich luźny stosunek do struktur władzy.

Pracownicy zespołowi to role, które łatwo dopasowują się do reszty zespołu, ewentualne niedogodności mogą być związane z wypadkową efektywnością. Najlepiej współpracują oni z innymi Pracownikami zespołowymi na zasadzie wzajemnego wsparcia. Mankamentem takiej relacji jest niezdecydowanie charakteryzujące tę rolę. Mogą oni również pomagać Filaram opracowywać pomysły. Najtrudniej Pracownikom zespołowymi współpracuje się z Animatorami, ponieważ Ci ostatni mogą zdeorganizować i zdominować relacje, dlatego też należy dbać o zabezpieczenie przed nadmierną dominacją ze strony Animatorów.

Gdy Animator jest szefem to najlepiej współpracuje mu się z Pracownikami zespołowymi, ponieważ Pracownik zespołowy radzi sobie z dominacją Animatora, choć występuje tu pewne ryzyko napięć. Animator dobrze nadzoruje również Realizatora pod warunkiem, że podwładny nie jest przyzwyczajony do stylu poprzednich szefów.

Dla Specjalisty pożądanym przełożonym jest taki, który: oferuje swobodę, wierzy w profesjonalizm i docenia. Właściwości te charakteryzują szefów Realizatorów. Trudniej układają im się natomiast relacje z kierownikami o typie Animatorów, ponieważ Ci ostatni naruszają terytorium Specjalisty jakim jest właśnie specjalizacja. Najlepszymi współpracownikami są natomiast Realizatorzy i Pracownicy zespołowi ze względu na łatwość stworzenia atmosfery wzajemnego szacunku. Najtrudniej jest natomiast z Filarami, ponieważ trudny teren stanowi styk pomysłowości z doświadczeniem.

Zaradne dusze jako przełożeni czerpią najwięcej satysfakcji z kierowania Finalizatorami, bo kompensują oni wszelkie mankamenty zachowań tego typu szefów, wiedząc jak postępować.

Finalizatorzy najlepiej wywiązują się z

zadań otrzymanych między innymi od Zaradnych dusz, czy Animatorów. W gronie współpracowników najlepiej czują się z Realizatorami ze względu na podobny styl i wartości.

Filary mają dobre równorzędne relacje na przykład z Pracownikami zespołowymi, natomiast najtrudniej z Realizatorami, ponieważ ich wartości w znacznym stopniu wykluczają się, a metody pracy są różne. Najmniej tolerancyjnymi dla Filara są szefowie Realizatorzy oraz Animatorzy, stąd Filar może przechodzić do stanowisk związanych z konsultingiem, a nie piąć się w górę.

Analogicznie, jak dla warstwy zadaniowej, można wymienić cechy pracownika, których występowanie świadczy o satysfakcjonującej, z punktu widzenia przełożonego, adaptacji w warstwie społecznej:

- „Traktuje współpracę jako możliwość podniesienia efektywności realizowanych zadań, a nie element rywalizacji
- Chętnie pomaga innym, kiedy tego potrzebują
- Potrafi poprosić innych o pomoc, gdy sobie z czymś nie radzi
- Podejmuje współpracę według zasady „wygrana-wygrana”
- Zarówno w stosunku do przełożonego jak i współpracowników wyraża szacunek i życzliwość
- Prezentuje postawę wyrażającą optymizm i pozytywne nastawienie
- Otwarty jest na uwagi i sugestie doskonalące jego kompetencje
- Swoją postawą wzbudza zaufanie innych” [Lista stanowi cytowanie oczekiwań pracodawców wymienionych w rozdz. 5 [Prze11].]

Podsumowanie

Artykuł starał się uświadomić, że nie należy skupiać się tylko na własnych zadaniach, a interakcji między współpracownikami ograniczać do minimum, bo wtedy nie można [Cant04]: zrealizować niektórych atrybutów jakości produktów, dla których pewne defekty uwidaczniają się dopiero po połączeniu zadań. Ponadto przy niskim poziomie komunikacji [Cant04] pracownik ma utrudnioną adaptację do środowiska pracy. Z drugiej strony nadmierna komunikacja, nie oparta na pewnych zasadach, jak na przykład plan komunikacji projektu prowadzi do chaosu. ■

Literatura

[Bath11] Bath G. i inni: Certified Tester. Expert Level Syllabus. Test Management (Managing Testing, Testers, and Test Stakeholders). Version 2011, ISTQB® 1 November 2011.

[BEL12] BELBIN® Associates. www.belbin.com 2012.

[Belb03] Belbin R. M.: Twoja rola w zespole, Gdańskie Wydawnictwo Psychologiczne. Gdańsk 2003.

[CaAh10] Capretz L.F. and Ahmed F. Making Sense of Software Development and Personality Types, IEEE IT Professional, Volume 12, Number 1, pp. 6-13, DOI: 10.1109/MITP.2010.33, IEEE Press, January-February 2010.

[Cant04] Cantor M.: Jak kierować zespołem programistów, WNT. Warszawa 2004.

[Capr11] Capretz L. F.: Software Testing & Human Factors, University of Western Ontario. Canada 2011.

[Fle05] Flannes S. W., Levin G.: Essential People Skills for Project Managers. Management Concepts, Inc. 1 sierpień 2005.

[MPM11] MBTI and Project Management, Chief Learning Officer – Middle East. <http://www.clo-me.com/?p=486> 24 maj 2011.

[NSE11] Xiong J.: New Software Engineering Paradigm Based on Complexity Science, Springer 2011.

[ONO12] O*NET OnLine <http://www.onetonline.org/> Luty 2012.

