

# Kształcenie inżynierii wymagań i procesy inżynierii wymagań według IREB

Włodzimierz Dąbrowski<sup>2</sup>, Andrzej Stasiak<sup>1</sup>, Krzysztof Wnuk<sup>3</sup>

1. Wojskowa Akademia Techniczna, Wydział Cybernetyki,  
ul. Kaliskiego 2, 00-908 Warszawa, [astasiak@wat.edu.pl](mailto:astasiak@wat.edu.pl)
2. Politechnika Warszawska, Instytut Sterowania i Elektroniki Przemysłowej  
ul. Koszykowa 75, 00-662 Warszawa, [w.dabrowski@ee.pw.edu.pl](mailto:w.dabrowski@ee.pw.edu.pl)
3. Uniwersytet w Lund, Wydział Informatyki,  
Ole Römers väg 3, 223 63 Lund, [wnuk@cs.lth.se](mailto:wnuk@cs.lth.se)

## Streszczenie

Inżynieria wymagań jest ważnym działem inżynierii oprogramowania. Znaczenie inżynierii wymagań i prawidłowego przebiegu procesów zarządzania wymaganiami jest często kluczowym czynnikiem sukcesu dla projektów informatycznych. Znane są przypadki, gdzie porażka projektu wynikała z błędów podczas analizy wymagań. [1, 2, 3, 4] Świadomość i znajomość tych procesów dla współczesnych inżynierów informatyków nie zawsze jest wystarczająca. W artykule został dokonany przegląd nauczania inżynierii wymagań w programach nauczania wybranych uczelni wyższych kształcących studentów na kierunku informatyka oraz metody kształcenia w tym zakresie. W ostatnich latach obserwuje się rozwój edukacji inżynierii wymagań nie tylko na wyższych uczelniach, ale również w kształceniu ustawicznym. Powstają nowe inicjatywy i organizacje szerzące wiedzę i dobre praktyki inżynierii wymagań. Na rynku pojawiają się też certyfikaty z tego obszaru wiedzy. W artykule przedstawiono jedną z takich organizacji - International Requirements Engineering Board (IREB) oraz jej podejście do procesów inżynierii wymagań i certyfikacji dostępnej od 2012 roku również w Polsce, a certyfikat IREB CPRET ma około 11 000 osób [18]

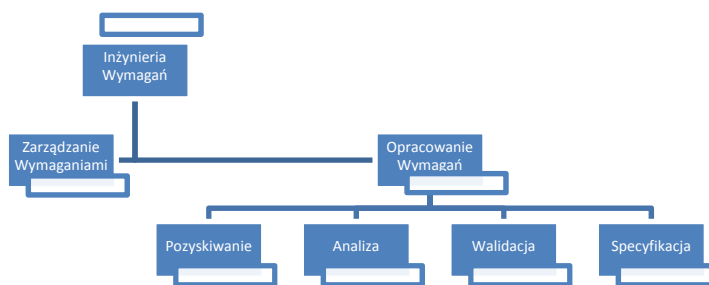
KEYWORDS: inżynieria wymagań, certyfikacja, IREB, kształcenie

## 1 Wprowadzenie

Inżynieria wymagań oprogramowania jest relatywnie nowym pojęciem w informatyce. Zostało ono stworzone w celu określenia wszystkich czynności związanych z szeroko rozumianym pozyskiwaniem, dokumentowaniem i zarządzaniem wymaganiami dla systemów komputerowych. Słowo „inżynieria” w tym kontekście należy rozumieć jako systematyczną i zdefiniowaną działalność zapewniającą powodzenie tych procesów. Widoczne jest tu zatem odmienne do tradycyjnego podejścia, w którym inżynieria rozumiana jest jako „*używanie właściwości materii, energii oraz obiektów abstrakcyjnych dla tworzenia konstrukcji, maszyn i produktów, przeznaczonych do wykonywania określonych funkcji lub rozwiązania określonego problemu*” [5].

W tradycyjnych podejściach do prowadzenia projektów informatycznych, takich jak V-model czy model kaskadowy zakłada się, że wszystkie wymagania zostaną zdefiniowane, a cały proces zakończy się, przed fazą projektowania i implementacji. Wymaga się, aby wszystkie wymagania zostały zdefiniowane i odpowiednio dobrze opisane. Nie dopuszcza się możliwości zmiany wymagań po ich pierwotnym ustaleniu, choć na przykład Royce dopuszcza iteracje i podkreśla rolę sprzężenia zwrotnego w procesie wytwarzania w modelu kaskadowym [19].

Metodyki zwinne określane także jako lekkie, zakładają inne podejście do pozyskiwania wymagań. W projektach prowadzonych zgodnie z ich wytycznymi dopuszcza się zmiany w trakcie trwania projektu jako jego naturalny element. W początkowej fazie definiowane się wymagania zgodne z obecnym stanem wiedzy. W czasie trwania projektu mogą one ulec zmianie, co kontrolowane jest przez wyznaczonych do tego członków zespołu.



Rys. 1.1 Podział dziedzinowy inżynierii oprogramowania, [5]

Zwinne wytwarzanie oprogramowania oparte jest o zasadę iteracyjnego (przyrostowego) wytwarzania oprogramowania. Kolejne etapy procesu zamknięte są w iteracjach, w których za każdym razem przeprowadza się testowanie wytworzonego kodu, zebranie wymagań, planowanie rozwiązań itd. W odróżnieniu od tradycyjnego modelu kaskadowego w podejściach zwinnych nie wymaga się zdefiniowania wszystkich wymagań na samym początku trwania projektu. Do pierwszej iteracji przystępuje się z definicją najważniejszych oczekiwań klienta w ramach następnych iteracji definiowane są kolejne wymagania, oraz następuje ich stopniowa implementacja i jak najszybsze zaimplementowanie „najmniejszej sprzedawalnej jednostki” (ang. smallest saleable unit), czyli minimum systemu które można sprzedać klientowi.

Metody zwinne (na przykład Scrum [20]) nastawiona są wytwarzanie oprogramowania wysokiej jakości oraz na jak najszybsze przedstawienie klientowi działającego prototypu produktu. W projekcie następują częste inspekcje wymagań i rozwiązań włącznie z procesem adaptacji. Pozawala to na wczesne wykrywanie błędów. Cechą charakterystyczną dla metodyk zwinnych jest angażowanie klienta w proces powstawania oprogramowania. Dzięki aktywnemu uczestnictwu interesariuszy, polegającym przede wszystkim na definicji wymagań i testowaniu przygotowanych rozwiązań, możliwe jest natychmiastowe reagowania na zaistniałe nieprawidłowości.

Metodyka ta najczęściej znajduje zastosowanie w małych zespołach programistycznych, w których nie występuje problem komunikacji, przez co nie trzeba tworzyć rozbudowanej dokumentacji kodu.

Najistotniejszym pojęciem używanym w inżynierii wymagań jest „wymaganie”. Według słownika języka polskiego „wymaganie jest to zespół warunków, które ktoś lub coś musi spełniać, jako synonimy tego pojęcia wymieniając słowa: pretensja, żądanie, oczekiwanie” [10]. W słowniku poświęconym terminom z dziedziny inżynierii oprogramowania można przeczytać że, wymaganie to:

- (1) Stan lub zdolność potrzebna użytkownikowi w celu rozwiązania problemu lub osiągnięcia celów.
- (2) Stan lub zdolność, które muszą być spełnione przez system lub element, tego systemu dla spełnienia umowy, standardu, specyfikacji, lub innej formalnie nałożonej dokumentacji.
- (3) Udokumentowane przedstawienie warunków lub zdolność opisanych w (1) i (2) [8].

Dean Leffingwell w swojej publikacji określa wymaganie jako: „możliwość rozwiązanie problemu i osiągnięcia celu, wymaganego przez użytkownika” [4], a także „możliwość spełnienia umowy, normy specyfikacji lub innej narzuconej dokumentacji, która musi mieć system lub komponent” [4].

Dla łatwiejszego zarządzania wymaganiami i odróżnienia ich ze względu na ich istotę dokonuje się podziału wymagań. Najczęściej spotykanym w literaturze podziałem jest ten na wymagania funkcjonalne i wymagania niefunkcjonalne, określane także jako jakościowe [3].

Wymagania funkcjonalne dotyczą tego co ma realizować system: jakie ma spełniać funkcje, jakich dostarczać usług, jak zachowywać się w określonych sytuacjach.

Ograniczenia dotyczące tego, jak system powinien realizować swoje zadania; np. wymagania dotyczące koniecznych zasobów, współpracy z określonymi narzędziami i środowiskami, zgodności z normami i standardami, a także przepisami prawnymi, w tym dotyczącymi tajności i prywatności. Wymagania niefunkcjonalne można dalej poddać dodatkowej klasyfikacji, w zależności od kategorii jakiej dotyczą. Bardzo powszechny jest podział FURPS. Pierwsza litera oznacza wymagania funkcjonalne, natomiast pozostałe - wymagania poza funkcjonalne.

U – użyteczność (ang. usability), oznacza łatwość użytkowania systemu. Takie wymagania można precyzować np. poprzez maksymalny czas szkolenia pracownika, liczba kontaktów ze wsparciem klienta, liczbą sytuacji, w których konieczne jest skorzystanie z systemu pomocy.

R – niezawodność (ang. reliability), może być mierzona poprzez: średnią liczbę godzin pracy bez awarii (ang. MTBF - Mean Time Between Failure), maksymalną liczbą godzin w miesiącu w ciągu których system może być wyłączony w celach pielęgnacyjnych (ang. maintenance) - ma to znaczenie szczególnie w przypadku systemów, które muszą pracować na okrągło - np. systemy bankowości elektronicznej

P – wydajność (ang. performance) - mierzona np. liczbą transakcji, którą system jest w stanie obsłużyć w ciągu godziny, liczbą użytkowników, którzy mogą być zalogowani jednocześnie do portalu.

S – bezpieczeństwo (ang. security), to wymagania związane z szyfrowaniem, polityką praw, itp. [8].

Często spotykanym rodzajem wymagań wyodrębnionym z wymagań niefunkcjonalnych są ograniczenia, które określają granice rozwiązania (ang. constraints). Niezależnie od tego jak problem jest rozwiązany, ograniczenia muszą być respektowane. Dla przykładu Amerykańskie Stowarzyszenie Inżynierów Kolei publikuje najlepsze praktyki dotyczące budowy systemów w kolejnictwie. Systemy produkowane dla kolejnictwa (również systemy IT) muszą być zgodne z określoną normą. [13]

## 2 Kształcenie w obszarze inżynierii wymagań

### Kształcenie na uczelniach polskich

Przegląd programów nauczania polskich uczelni technicznych pod kątem kształcenia w obszarze inżynierii wymagań wskazuje, że zarówno zakres, jak i intensywność edukacji w dziedzinie inżynierii w znacznym stopniu zależy od: poziomu edukacji (stopnia studiów), jak i specjalizacji na kierunku informatyka. W większości uczelni inżynieria wymagań nauczana jest w ramach szerszych zagadnień inżynierii oprogramowania lub analizy i modelowania systemów. W ramach badań autorzy dokonali przeglądu metod nauczania inżynierii wymagań na czterech uczelniach: Politechnice Warszawskiej, Wojskowej Akademii Technicznej, Polsko-Japońskiej Wyższej Szkole Technik Informatycznych i Wyższej Szkole Informatyki i Zarządzania.

Zagadnienia inżynierii wymagań zasadniczo prezentowane są na studiach I stopnia. Na studiach drugiego stopnia przyjmuje się na ogół założenie pełnej znajomości zagadnień związanych z inżynierią wymagań wynikającą z dwóch obowiązujących obecnie nurtów w zakresie procesów wytwórczych, a mianowicie zwinnej (agile – zwykle w wydaniu SCRUM) i tradycyjnej (czy tzw. klasycznej – reprezentowanej przez RUP) inżynierii wymagań. Warto tu również zaznaczyć, że zasadniczym sposobem weryfikacji wiedzy i umiejętności w tym zakresie jest realizacja projektów zespołowych, które znacznie wykraczają swoją złożonością poza "tzw. projekty akademickie" – angażując zasoby niejednokrotnie przekraczające poj. grupę studencką, tj. których liczność jest większa niż 20.

Studenci studiów I stopnia w procesach pozyskiwania wymagań w ramach grup projektowych wyodrębniają ze swego składu interesariuszy, reprezentujących zamawiającego projekt i przygotowują wspólnie z wykładowcą dokumenty definicji projektu, które następnie zespół projektowy przekształca w artefakty etapu rozpoczęcia projektu (zgodnie z procesem wytwórczym RUP). Ten krok jest ważny, ponieważ dokumenty te są poddawane analizie lingwistycznej przez zespół projektowy na podstawie, której ustaleni są kandydaci na obiekty projektowe (w pierwszej kolejności stanowią one kandydatów na pojęcia słownikowe, a następnie po odpowiedniej filtracji grup rzeczownikowych i czasownikowych stanowią kandydatów na: aktorów, klasy, atrybuty oraz przypadki użycia i operacje). W ramach projektu studenci, dla uzyskania akceptacji opracowanych przez zespół projektowy scenariuszy, wykorzystują sesje CRC (zwykle tylko dla kluczowych przypadków użycia systemu) odgrywając „role” obiektów projektowych zgodnie ze zidentyfikowanym scenariuszem). Warto tu zaznaczyć, że obie te techniki: mechanistyczna analiza lingwistyczna i systematyczna, jako sesje CRC, są stosowane przez zespoły realizujące procesy inżynierii wymagań zarówno zwinnie, jak i klasycznie.

Ze względu na dużą licznosc zespołów projektowych dużą wagę przywiązujemy do narzędziowego wsparcia procesów gromadzenia, zarządzania i śledzenia wymagań przez zespoły projektowe. W tym zakresie wykorzystujemy zarówno środowiska desktopowe (do pracy lokalnej) jak i serwerowe, dostępne przez platformę WEB. Nauka narzędzi realizowana jest w formie samokształcenia na podstawie przygotowanych materiałów szkoleniowych i praktycznie weryfikowana jest podczas warsztatów i przeglądów projektu. Warto tu również zaznaczyć, że każdy uczestnik projektu nie tylko uczestniczy w jego realizacji, ale również mierzy swój wysi-

lek, korzystając z metryk wspieranych przez platformy narzędziowe. Dzięki temu, jako wykładowcy, mamy szansę zobiektywizowanej oceny nakładów pracy poniesionych przez poszczególnych studentów jako członków zespołów projektowych. Aby osiągnąć cele realizowanych przedmiotów, zespoły część wymagań modelują:

- 1) zwinnie: dla metodyki SCRUM tworząc opisy sytuacji (user story), artefakty w postaci wymaganych planów, jako zaległości produktowe, wersji i sprintu poznając odpowiednie praktyki: spotkania produktowe, sprintu, retrospektywne;
- 2) zgodnie z procesem RUP: w oparciu o model FURPS, tworzą: wymagania słownikowe - TERM, żądania udziałowca - STRQ, cechy systemu – FEAT, wymagania typu przypadki użycia – UC, wymagania poza funkcjonalne (dodatkowe) – SR;

Istotne w procesie edukacji inżynierii wymagań jest również nauczanie dokumentowania wymagań. Dlatego studenci opracowują specyfikacje wymagań w oparciu o standardowe wzorce: formalny, nieformalny, tabelaryczny, czy RUP. Zwykle najwięcej trudności przysparza studentom opracowanie poprawnego modelu wymagań wyrażonego w języku UML, a w nim: procesy strukturalizacji wymagań oraz zarządzanie ich złożonością.

Z tego względu szczególnie istotne jest dokonanie przeglądu wymagań, najlepiej w oparciu o jednolite kryteria (Tabela. 1).

Tabela 1. Formularz oceny wymagań opracowanych zgodnie z procesem wytwórczym RUP.

Lp	Artefakty oceniane podczas przeglądu	Max.
1	Dokument: przedstawienie problemu - dokument wykonawcy	5
2	Dokument: wizja projektu Dyplom	5
3	Raport: wyniki analizy lingwistycznej - lista kandydatów na usługi (przypadki użycia) i klasy systemu	5
4	Dokument: Słownik systemu Dyplom – z pojęciami w notacji BNF.	5
5	Opracowanie planu (zaległości produktowych i sprintu)	5
6	Raport: wymagania typu STRQ (żądania udziałowców)	5
7	Raport: wymagania typu TERM (słownikowe)	5
8	Raport: wymagania typu FEAT (cechy systemu)	5
9	Raport: wymagania typu: UC (wymagania funkcjonalne)	5
10	Raport: widok śledzenia żądań udziałowców na cechy systemu	5
11	Raport: model przypadków użycia	5
12	Raport: modele aktywności (dla wybranych przypadków użycia)	5
13	Raport: Definicje kart CRC + przykładowe scenariusze	5
14	Tablice: jako widoki realizacji planów, przez członków zespołu, na platformie jazz	5
<b>Razem:</b>		<b>70</b>

Drugi stopień edukacji w zakresie inżynierii wymagań, zasadniczo związany jest z formalizacją wymagań oraz wprowadzeniem specjalizowanych procesów wytwórczych (np.: SOMA) – dla zachowania spójności w programach nauczania<sup>1</sup> oraz standaryzacją koncepcji architektonicznych (od MDA, do MDSD) - ze względu na specjalizację kierunków nauczania (sieci komputerowe, multimedia, systemy wbudowane, bezpieczeństwo SI) niezbędne jest rozszerzenie zagadnień z inżynierii oprogramowania o zagadnienia dotyczące inżynierii systemów (ang. Systems Engineering).

W bieżącym roku przeprowadziliśmy eksperyment, którego wyniki są zadziwiająco pozytywne, polegający na realizacji przez studentów II stopnia podczas zajęć laboratoryjnych środowiska do modelowej weryfikacji semantyki modelu wymagań opracowanego przez studentów stopnia I, wykorzystując opracowany przez siebie DSL (i odpowiadający mu – zdefiniowany przez studentów profil UML2).

<sup>1</sup>) w ramach studiów I stopnia studenci zwykle budują złożone rozwiązania portalowe, natomiast w ramach drugiego stopnia rozbudowują (bądź tylko modyfikują) je do modelu usługowego, budując rozwiązanie w architekturze SOA.

<sup>2</sup>) Projekt dotyczył walidacji wymagań, wobec systemu wspomagania obsługi egzaminów dyplomowych na uczelni wyższej. Weryfikacja narzędzia polegała na zapisaniu przez wszystkich członków zespołu w utworzonym profilu modelu swojego egzaminu dyplomowego (na studiach I stopnia).

## Kształcenie na uczelniach szwedzkich

System kształcenia inżynierii oprogramowania i inżynierii wymagań w Szwecji oparty jest na dwóch celach edukacyjnych: po pierwsze na połączeniu teorii z praktyką poprzez użycie wykładanych procesów, technik i metod w projektach i po drugie na pracy grupowej w formie projektu studenckiego. Oba te założenia powodują przesunięcie ciężaru edukacyjnego w stronę projektu i zminimalizowanie ilości materiału wykładanego na rzecz samodzielnej edukacji w oparciu o materiały z kursów. Wszystkie kursy z inżynierii oprogramowania oferowane na uniwersytecie w Lund są oparte o projekt studencki w grupach do sześciu studentów. Dzięki położeniu nacisku na projekty, studenci doświadczają typowych problemów w projekcie jak problemy z komunikacją, nierówne zaangażowanie członków zespołu, brak motywacji oraz nieprzewidziane sytuacje. Każdy z projektów zakończony jest raportem, w którym studenci opisują swoje doświadczenia oraz analizują co mogliby zrobić inaczej, a co zrobili dokładnie tak samo.

Kursy na szwedzkich uczelniach są intensywniejsze w stosunku do polskiego systemu. Rok akademicki dzieli się na cztery „okresy nauczania”, każdy zakończony tygodniową sesją egzaminacyjną. W każdym okresie, studenci intensywnie studiują niewielką ilość przedmiotów. Dla przykładu, kurs inżynierii wymagań oferowany przez Uniwersytet w Lund [14] jako odrębny kurs na poziomie zaawansowanym trwa siedem tygodni: zawierając w sobie sześć wykładów po dwie godziny, pięć ćwiczeń po dwie godziny i projekt, którego oszacowany wkład każdego członka zespołu wynosi około trzy tygodnie pracy. W rezultacie Szwedzcy studenci w większości studiują dwa kursy po 7.5 punktów ECTS jednocześnie. Większa liczba kursów stanowi poważne obciążenie czasowe. Intensyfikacja przebiegu kursów wraz ze zmniejszeniem ich ilości minimalizuje liczbę równoległych tematów studiowanych przez studentów. Dzięki temu mogą oni zagłębić się w szczegóły zagadnień poruszanych na kursach i w czasie projektu. Poprzez większą ilość zajęć w tygodniu, studenci częściej powracają do wykładanego materiału i dzięki temu prawdopodobieństwo zapamiętania go na dłużej wzrasta.

Cele kursu inżynierii wymagań na uniwersytecie w Lund są podzielone na trzy kategorie. Do pierwszej kategorii zalicza się cele związane z pozyskaniem odpowiedniej wiedzy w inżynierii wymagań. W tym przypadku studenci kończący kurs powinni rozumieć podstawowe cele i zakorzenione problemy inżynierii wymagań dla systemów informatycznych. Powinni oni rozumieć role interesariuszy w procesie inżynierii wymagań oraz poprawnie zdefiniować typ projektu w którym pracują (kontrakt lub produkt przeznaczony dla danego rynku). Studenci powinni rozumieć pojęcie wymagań funkcjonalnych oraz umieć rozróżnić wymagania funkcjonalne od нефункциональных oraz zdefiniować je dla danego projektu. Studenci powinni również rozumieć role dokumentacji wymagań oraz specyfikacji wymagań, włączając w to najważniejsze kryteria jakościowe dobrej specyfikacji wymagań. Wymaga się również od studentów zrozumienia różnicy pomiędzy inżynierią wymagań oraz procesem projektowania oprogramowania jak również podstawowej wiedzy na temat narzędzi używanych w inżynierii wymagań oraz praktyki w przemyśle w oparciu o podaną literaturę empiryczną.

Do celów powiązanych ze zdobyciem określonych umiejętności poprzez uczestnictwo w kursie zaliczone zostały: umiejętność skutecznego definiowania, dokumentowania, walidacji oraz nadawania priorytetów wymaganiom. Dodatkowo kurs zawiera naukę umiejętności używania kilku technik do wyżej wymienionych zadań w zależności od rodzaju kontekstu i projektu oraz umiejętności określania jakości już istniejących specyfikacji wymagań. Kurs uczy również umiejętności czytania ze zrozumieniem artykułów naukowych z dziedziny inżynierii wymagań. Studenci otrzymują osiem artykułów do przeczytania i przedyskutowania w ramach kursu.

Do celów powiązanych z ostatnią kategorią zalicza się cele związane z przekazaniem odpowiedniego nastawienia studentów do zagadnień inżynierii wymagań. W tym przypadku przekazuje się studentom opinie, że lepiej jest zrobić coś dobrze od początku niż później poprawiać oraz przekazuje się opinie, że wysoka jakość wymagań pozytywnie wpływa na wysoką jakość produktu. Kurs stara się również przekazać, że użyteczność oraz dopasowanie metody do kontekstu są bardzo ważne w inżynierii wymagań oraz, że należy zawsze brać pod uwagę czynnik ludzki. Dodatkowo, przekazywane jest przeświadczenie, że inżynier wymagań powinien pomagać interesariuszom w dążeniu do zdefiniowania realistycznego produktu który można wytworzyć w ramach podanego budżetu.

Kurs inżynierii wymagań umiejscowiony jest na poziomie zaawansowanym (w dniu publikacji tej pracy wydział informatyki uniwersytetu w Lund oferował tylko jednostopniowe studia magisterskie zakończone tytułem magistra inżynieria cywilnego ze specjalizacją w jednej z poniższych dziedzin). Kurs jest kursem obieralnym i obejmuje studentów kierunków programu inżynierii danych, ekonomii przemysłowej, elektroniki oraz

telekomunikacji i komunikacji danych. Kurs jest również otwarty dla studentów zagranicznych, osób pracujących w przemyśle oraz studentów innych kierunków na uniwersytecie w Lund. W przypadku słuchaczy pracujących w przemyśle posiadających duże praktyczne doświadczenie w pracy jako inżynierowie wymagań, projekt kursu jest indywidualnie dopasowywany do potrzeb i zainteresowań słuchaczy. Często temat projektu jest bezpośrednio powiązany z wyzwaniami z którymi zmierzają się słuchacze pracujący w przemyśle. Certyfikat IREB nie jest oferowany dla studentów kursy inżynierii wymagań ze względu na rozbieżności pomiędzy zakresem kursu, w szczególności w zakresie inżynierii wymagań dla produktów przeznaczonych na otwarty rynek a zakresem podstawowego kursu certyfikatu IREB.

Uniwersytet w Lund nie umożliwia kształcenia inżynierów wymagań. Studenci mogą jednak wybrać temat pracy dyplomowej z zakresu inżynierii wymagań. Prace te często odbywają się w ścisłej współpracy z przemysłem i skupiają się na zrozumieniu autentycznych sytuacji i wyzwań inżynierii oprogramowania jak również zaproponowanie i sprawdzenie rozwiązań do tych problemów. Do firm w których prace dyplomowe się odbyły lub obecnie odbywają zaliczmy: Sony Mobile, ST Ericsson, Ericsson, ABB, Anoto, Axis Communications oraz wiele firm doradczych. Ze względu na stopień zaawansowania projektów z przemysłem a co za tym idzie wymagany większy wkład pracy niż budżet kursu inżynierii wymagań, projekty z przemysłem realizowane są głównie w oparciu o prace magisterskie w wymiarze dwudziestu tygodni pracy. Podczas pracy magisterskiej w przemyśle, studenci są opłacani zgodnie z obowiązującym wynagrodzeniem dla studentów oraz otrzymują miejsce pracy, dostęp do zasobów firmy i bardzo często szybko integrują się z nowym środowiskiem. Dzięki pomocy doświadczonych nauczycieli prace dyplomowe zawierają użyteczne dla przemysłu analizy lub rozwiązania jak i prezentują wymaganą oryginalność naukową. Większość prac magisterskich jest po obronie publikowana jako artykuł na konferencji lub w czasopiśmie, np. Information and Software Technology (IST).

W ramach kursu inżynierii wymagań studenci poznają i pracują z narzędziami otwartymi jak i dostępnymi za licencją. Z narzędzi dostępnych w ramach licencji studenci mogą używać IBM Rational Focal Point do ustalenia priorytetów wymagań oraz IBM Rational Doors [21] do dokumentacji wymagań. Dodatkowo wprowadzone są do kursu dwa narzędzia stworzone przez ośrodki badawcze w Lund i Calgary. Studenci mogą użyć prototypowego narzędzia reqT [15] do modelowania wymagań tekstowych. ReqT napisany jest w języku Scala i umożliwia tworzenie i zarządzanie modelami wymagań wykorzystując uniwersalne kolekcje, połączenie języka modelowania i bezpiecznego typu. ReqT umożliwia również współpracę z aplikacjami arkusza kalkulacyjnego oraz publikowanie wymagań w Internecie lub jako dokumenty PDF. ReqT oferuje rozszerzenie otwartego, wewnętrznego DSL (języka specyficznej domeny) o własne semantyki modelowania. Drugim narzędziem stworzonym w ośrodku badawczym w Calgary jest Release Planner stworzony przez ekspertów w zakresie planowania kilku wersji oprogramowania [16]. Release planner używa algorytmów ogólnych do zoptymalizowania ilości i zakresu wymagań które powinny być zawarte w każdej z wersji oprogramowania i umożliwia analizę „co by było gdyby” na przykład co by były gdyby jakieś wymaganie zostało usunięte, jak zmienia to ogólne plany i czy powoduje jakieś opóźnienia. Studenci używają tego narzędzia do określenia zakresu kilku wersji oprogramowania w oparciu o określone wymagania.

### 3 International Requirements Engineering Board

IREB (*International Requirements Engineering Board*), czyli Międzynarodowa Rada Inżynierii Wymagań powstała w 2006 roku w Fürth w Niemczech. IREB stworzyła i nadzoruje międzynarodowy system akredytacji i certyfikacji w dziedzinie inżynierii wymagań, zwany CPRE (Certyfikowany Specjalista Inżynierii Wymagań (*Certified Professional in Requirements Engineering*)). Celem IREB jest stworzenie i dalsza budowa jednolitego, powszechnego, międzynarodowego systemu kwalifikacji zawodowych dla inżynierów wymagań [12].

Jako podstawowe objawy złego podejścia do inżynierii wymagań IREB określa przede wszystkim brak lub niedoprecyzowanie wymagań [1, 2, 3, 4]. Za typowe przyczyny takiego stanu rzeczy uznaje pochopne przyjmowanie pewnych założeń co do definicji systemu za pewne i z góry oczywiste, uznając że nie ma potrzeby konsultacji tych aspektów między wykonawcą systemu a zamawiającym. Problemem są również zbyt wygórowane lub ambitne oczekiwania klientów.

Zgodnie z podejściem IREB istnieją cztery główne działania związane z wymaganiami, które mają kluczowe znaczenie w procesie wytwarzania oprogramowania. Tymi czterema filarami są:

- pozyskiwanie wymagań,
- dokumentacja,
- walidacja i negocjowanie,
- zarządzanie wymaganiami.

### Pozyskiwanie wymagań

Jako podstawę do pozyskania wymagań możemy traktować określony kontekst systemu, w ramach którego działają inżynierowie wymagań. Określa on domenę działania oraz definiuje źródła wymagań. Za źródła wymagań uznać można wszystkie osoby bezpośrednio lub pośrednio zainteresowane w powodzenie przedsięwzięcia jakim jest tworzenie nowego systemu informatycznego, a także wszelkie rodzaje artefaktów pojawiające się w tym procesie. W świetle tej definicji za źródła wymagań uznać możemy np. udziałowców, dostępna dokumentacja czy tak zwane systemy spadkowe. Obecnie ponad 80% obecnych systemów to systemy oparte o rozwinięcie już istniejącego rozwiązania.

Udziałowcy, jako najważniejsze źródło wymagań, muszą być obdarzeni szczególną uwagą ze strony zespołu projektowego. Bardzo istotne jest ich aktywizowanie tak, aby brali oni czynny udział w projekcie, a nie byli tylko jego biernymi obserwatorami. W tym celu zgodnie z kulturą organizacyjną firmy, należy uzgodnić ustnie lub też w formie pisemnej, role i kompetencje udziałowców w całym procesie. Zabezpiecza to przed brakiem motywacji i konfliktami, mogącymi negatywnie wpłynąć na powodzenie przedsięwzięcia.

Istotnie jest, aby przy pozyskiwaniu wymagań można było poznać jak ważne dla zadowolenia interesariuszy są poszczególne wymagania. Pozwala to na ustalenie priorytetów wymagań tak, aby jak najbardziej spełnić oczekiwania udziałowców. Pomocny w tym procesie jest model Kano. Dzieli on wymagania na trzy grupy:

- czynniki podstawowe (synonim: wzbudzające niezadowolenie),
- czynniki wydajności (synonim: wzbudzające zadowolenie),
- czynniki entuzjazmu (synonim: wzbudzające zachwyty).

W celu odpowiedniego pozyskania (świadomych, nieświadomych oraz pozostałych) wymagań udziałowców stosuje się szereg technik zbierania wymagań. Istnieje wiele czynników definiujących wybór odpowiedniej techniki. IREB zaleca wzięcie pod uwagę m. in. czynniki ryzyka, wpływy osobowe, wpływy organizacyjne, wpływy funkcyjno-treściowe oraz zamierzony poziom szczegółowości wymagań. Dobór odpowiedniej techniki jest bardzo ważną umiejętnością i jest kluczowy dla powodzenia projektu. Dlatego należy poświęcić odpowiednio dużo czasu na wybór adekwatnej techniki, pamiętając o tym, że najlepsze efekty osiągnąć można przy ich łączeniu.

### Dokumentacja wymagań

W inżynierii wymagań konieczne jest dokumentowanie wszelkich istotnych informacji. Techniki dokumentacji są w zależności od przyjętego modelu bardziej lub mniej formalnymi sposobami reprezentacji wymagań, począwszy od zapisu w formie języka naturalnego po diagramy o zdefiniowanej semantyce. Ze względu na to, że nad pozyskiwaniem wymagań pracuje zazwyczaj wiele osób, istotna jest zapewnienie właściwej komunikacji między nimi.

Oddzielną kwestią pozostają formy dokumentacji wymagań. IREB proponuje trzy skuteczne metody w zależności od perspektywy, dla której opisujemy wymagania: (1) dokumentacja wymagań w języku naturalnym, (2) koncepcyjne modele wymagań takie jak diagramy przypadków użycia, diagramy klas, diagramy aktywności czy diagramy stanów, (3) mieszane formy dokumentacji wymagań.

Zgodnie z założeniami IREB, główną częścią dokumentu wymagań pozostają wymagania dla systemu. Poza samymi wymaganiami, w zależności od celu powstania dokumentacji, dokumenty wymagań zawierają także informacje na temat kontekstu systemu, warunków zatwierdzenia lub, przykładowo właściwości implementacji technicznej. Aby zapewnić możliwość zarządzania dokumentami wymagań, dokumenty te muszą mieć formalną strukturę. W tym zakresie IREB zaleca stosowanie referencyjnej struktury dokumentu zgodnej z IEEE-830 [22].

W praktyce okazuje się, że istnieje dużo pozytywnych skutków wykorzystywania struktur referencyjnych dla dokumentów wymagań. Przykładowo wykorzystanie struktur referencyjnych ułatwia korzystanie z dokumentów wymagań w późniejszych czynnościach konstrukcyjnych (np. w definiowaniu przypadków testowych). Na ogół struktury referencyjne nie mogą być stosowane dla dokumentu wymagań na zasadzie „jedna struktura dla jednego dokumentu”, jako że struktura treści konkretnego dokumentu wymagań musi być często szczegółowo dopasowywana do okoliczności właściwych dla domeny, firmy czy projektu.

## **Negocjowanie i walidacja wymagań**

Celem walidacji wymagań jest sprawdzenie czy wymagania spełniają zdefiniowane kryteria jakości (np. poprawność lub kompletność) tak, aby wykryć i skorygować błędy w wymaganiach tak wcześnie, jak to możliwe na etapie ich definicji.

Nierozwiązane konflikty w wymaganiach systemu oznaczają, że jakiś zbiór wymagań interesariuszy nie może być zrealizowany, lub, że działający system albo nie zostaje w ogóle przyjęty, nie będzie w ogóle używany, lub będzie używany w niedostatecznym stopniu. Celem negocjowania wymagań jest określenie wspólnego i uzgodnionego porozumienia, które pozwoli uniknąć takich sytuacji.

Drugim aspektem poruszonym równoległe z walidacją wymagań jest ich negocjowanie. Celem negocjowania wymagań jest ustalenie wspólnej dla wszystkich interesariuszy interpretacji wymagań wobec tworzonego systemu. Czynności wchodzące w skład negocjowania wymagań to:

- rozpoznanie konfliktów,
- analiza konfliktów,
- rozwiązywanie konfliktów,
- dokumentowanie rozwiązań konfliktów.

## **Zarządzanie wymaganiami**

Aby zarządzać wymaganiami systemowymi podczas całego cyklu życia, konieczne jest zebranie informacji o wymaganiach w formie uporządkowanych atrybutów. Aby zdefiniować strukturę atrybutów wymagań, stosuje się odpowiednie zestawienia, które przedstawia się albo w postaci tabeli, albo w postaci modelu informacji.

W praktyce projektowej liczba wymagań i zależności pomiędzy nimi nieustannie wzrasta. Żeby skutecznie zarządzać złożonymi wymaganiami, konieczna jest selekcja dostępu, co pozwala na dobór wymagań potrzebnych tylko do bieżącego zadania.

Bardzo istotne w kontekście określanie harmonogramu prac i istotności wymagań zamawiających, jest ustalenie priorytetów wymagań. Nadawanie priorytetów wymaganiom odbywa się w różnym czasie, podczas różnych zadań w projekcie i na podstawie różnych kryteriów. Przygotowanie do nadawania priorytetów odbywa się według następujących zasad: (1) określenie celów i ograniczeń procesu nadawania priorytetów, (2) określenie kryteriów nadawania priorytetów, (3) określenie istotnych interesariuszy, (4) wybranie wymagań lub innych wytworów, których priorytety mają być ustalone.

Wymagania zmieniają się podczas całego cyklu życia oprogramowania. Zmianami wymagań zarządza się i realizuje je w systematycznym, zdefiniowanym procesie zarządzania zmianami. W tym procesie rada kontroli zmian odpowiada za przetwarzanie przychodzących żądań zmian. Do zadań rady kontrolowania zmian należą: (1) klasyfikacja przychodzących żądań zmian, (2) określenie wysiłku potrzebnego do przeprowadzenia zmiany, (3) ocena relacji kosztu dla żądanej zmiany, (4) stworzenie nowych wymagań zgodnie z żądaniem zmiany, (5) decyzja czy zaakceptować, czy odrzucić zgłoszenie zmiany, (6) określenie priorytetu zaakceptowanego zgłoszenia zmiany, (7) przydzielenie zaakceptowanej zmiany do konkretnego projektu.

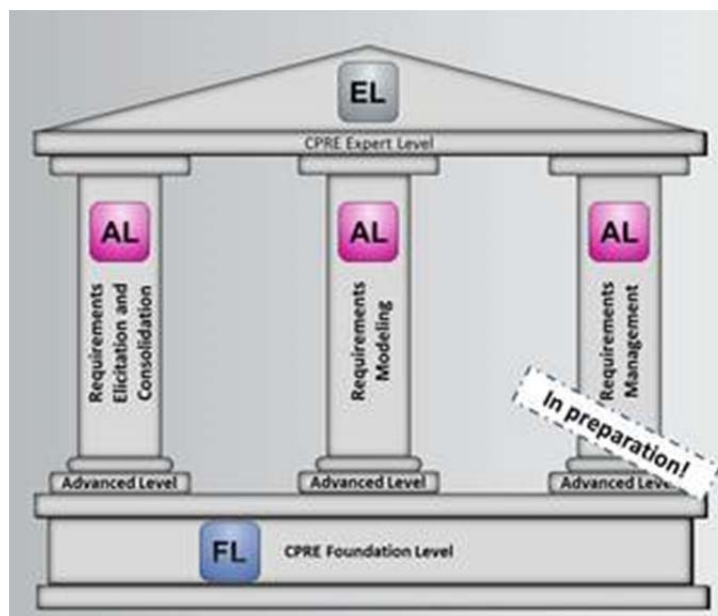
Członkowie rady kontrolowania zmian to zwykle kierownik zarządzania zmianami, zleceniodawca, architekt oprogramowania, przedstawiciel użytkowników, kierownik do spraw jakości oraz inżynier wymagań.

## **Certyfikacja IREB**

IREB stworzył i nadzoruje międzynarodowy system akredytacji i certyfikacji w dziedzinie inżynierii wymagań, zwany CPRE (Certyfikowany Specjalista Inżynierii Wymagań – *Certified Professional in Requirements Engineering*). Celem IREB jest stworzenie i dalsza budowa jednolitego, powszechnego, międzynarodowego systemu kwalifikacji zawodowych dla inżynierów wymagań.

Stworzony przez IREB system składa się z trzech poziomów: podstawowego, zaawansowanego i eksperckiego. W skład każdego poziomu wchodzi kompendium wiedzy (tzw. sylabus), zasady akredytacji oraz egzaminów certyfikacyjnych i pytania egzaminacyjne.





Rys. 1.2 Schemat certyfikacji IREB [18]

Poziom podstawowy dostępny jest w tej chwili po angielsku, francusku, niemiecku, hiszpańsku, portugalsku, po polsku i po szwedzku. IREB jest kierowana przez 12-osobowy Komitet Sterujący, w skład którego wchodzi przedstawiciele firm informatycznych, wyższych uczelni oraz innych organizacji. Inicjatywy IREB na forum międzynarodowym realizuje 35-osobowa Grupa Wspierająca.

Od 2007 roku ponad 11 000 fachowców pozytywnie przeszło egzaminy CPRE Poziomu Podstawowego – z tendencją wzrostową. Jak dotąd, większość egzaminów certyfikacyjnych odbywała się w Niemczech, Austrii i Szwajcarii, ale certyfikaty CPRE zdobyli już także fachowcy z Holandii, Hiszpanii, Bułgarii, Egiptu, USA, Kolumbii, Brazylii, Korei Południowej, Polski, Malesji oraz Indii.

W Polsce inicjatywa certyfikacji i szkoleń IREB rozwijana jest od roku 2011. Obecnie w Polsce działa dziewięćosobowa Rada IREB złożona ze specjalistów zajmujących się inżynierią wymagań z uczelni i przemysłu informatycznego. Do zadań Rady należy: przetłumaczenie słownika terminologii oraz kompendium podstawowego CPRE na język polski, przetłumaczenie i wydanie podręcznika do kursu podstawowego CPRE na język polski, tłumaczenie na język polski kompendiów zaawansowanych CPRE.

Podstawowy egzamin CPRE jest oferowany również studentom kończącym kursy inżynierii oprogramowania w dwu polskich uczelniach: Politechnice Warszawskiej oraz PJWSTK jako nieobowiązkowy element kształcenia kursu inżynierskiego.

## Podsumowanie

Kształcenie inżynierów w obszarze inżynierii wymagań jest istotnym elementem wykształcenia współczesnego inżyniera informatyka. Choć w większości wypadków uczelnie nie dedykują tym zagadnieniom odrębnego kursu, to jednak inżynieria wymagań jest istotnym elementem kształcenia we wszystkich programach nauczania dla kierunku informatyka. W większości wypadków kształcenie to odbywa się metodami projektowymi, co należy podkreślić z satysfakcją. W odróżnieniu od programów kształcenia na uczelniach szwedzkich w Polsce zdecydowanie rzadziej wykonywane są „rzeczywiste” projekty przy współpracy z przemysłem. Z pewnością wynika to z ogólnych ułomności polskiego systemu kształcenia, który w niewystarczającym stopniu powiązany jest jeszcze z problemami i potrzebami przemysłu informatycznego.

W badaniach rynku widoczne jest też zainteresowanie przemysłu szkoleniami z tego zakresu. Pojawienie się takich organizacji jak IREB zdaje się potwierdzać potrzebę kształcenia i certyfikacji w tym obszarze.

W polskich realiach jednak certyfikacja ta jest jeszcze na początku drogi. Zainteresowanie certyfikacją taką jak CPRE jest jak na razie ograniczone, ale biorąc pod uwagę trendy światowe powinno w najbliższym czasie wzrosnąć.

## Bibliografia

- [1] "New Standish Group report shows more projects failing and less successful projects" (2009). The Standish Group, [http://www.standishgroup.com/newsroom/chaos\\_2009.php](http://www.standishgroup.com/newsroom/chaos_2009.php)
- [2] "KPMG New Zealand Project Management Survey 2010" (2010). KPMG, <http://www.kpmg.com/NZ/en/IssuesAndInsights/ArticlesPublications/Documents/Project-Management-Survey-report.pdf>
- [3] "Flawed Requirements Trigger 70% of Project Failures" (2006). Info-Tech Research Group, <http://www.infotech.com/research/flawed-requirements-trigger-70-of-project-failures>
- [4] Galorath, Dan (2008). "Software Project Failure Costs Billions – Better Estimation & Planning Can Help", <http://www.galorath.com/wp/software-project-failure-costs-billions-better-estimation-planning-can-help.php>
- [5] Dean Leffingwell, Zarządzanie Wymaganiami, WNT, Warszawa 2003
- [6] Winston Royce, Managing the Development of Large Software Systes, Proceedings IEEE Wescon, August 1970, pp 1-9
- [6] Yan Sommerville & Pete Sawyer '97 Requaierments Eengeering Good Praktics, Pearson Education, 2003
- [8] IEEE Standards Coordinating Committee, IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610, 1990
- [12] [www.ireb.org.pl](http://www.ireb.org.pl),
- [13] <http://www.arena.org/publications/>
- [14] <http://cs.lth.se/ets170>
- [15] <http://www.reqt.org/>
- [16] [www.expertdecisions.com/](http://www.expertdecisions.com/)
- [18] [www.ireb.org](http://www.ireb.org)
- [19] W. W. Royce "Managing the Development of Large Software Systems", Proceedings, IEEE WESCON, August 1970, pages 1-9.
- [20] <http://www.scrum.org/>
- [21] <http://www-01.ibm.com/software/rational/>
- [22] <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=720574>