

Należyta staranność w inżynierii wymagań i modelowaniu systemów

Anna Bobkowska
Politechnika Gdańska

1. Wstęp

Wypracowanie schematów postępowania podczas stwierdzania, czy została dołożona należyta staranność podczas pozyskiwania i specyfikacji wymagań jest zagadnieniem interdyscyplinarnym na przecięciu prawa i inżynierii oprogramowania. Jednak pomiędzy inżynierią oprogramowania a prawem występują duże różnice w zakresie wiedzy, specjalistycznej terminologii, stosowanych podejść i metod, a także sposobów myślenia osób reprezentujących te dwie profesje. Poszukiwanie rozwiązań problemów na przecięciu dwóch dyscyplin wymaga porozumienia i współpracy pomiędzy przedstawicielami tych dwóch grup zawodowych.

W referacie podjęto próbę odpowiedzi na następujące pytania: Co o inżynierii wymagań i o analizie systemów powinien wiedzieć prawnik, aby mógł stwierdzić lub przeprowadzić proces stwierdzania, czy została zachowana należyta staranność podczas pozyskiwania wymagań i ich specyfikacji? Kto jest odpowiedzialny za problemy wynikające z nieprawidłowego zidentyfikowania wymagań? Jakie czynniki należy wziąć pod uwagę podczas oceny, czy została dołożona należyta staranność w zakresie pozyskiwania wymagań oraz modelowania i analizy systemu informatycznego?

W skład referatu wchodzi prezentacja następujących zagadnień:

- podstawowe pojęcia i zagadnienia inżynierii wymagań oraz problemy z wymaganiami istotne z punktu widzenia rozstrzygania na temat należytej staranności w zakresie inżynierii wymagań;
- podstawy modelowania i analizy systemów informatycznych, których zastosowanie jest sprawdzoną metodą na przewyższenie niektórych problemów z wymaganiami;
- wpływ jakości wymagań oraz jakości modeli analitycznych na jakość oprogramowania;
- należyta staranność podczas pozyskiwania, specyfikacji, analizy, weryfikacji i zarządzania zmianami wymagań oraz podczas modelowania i analizy systemów;
- perspektywa obiektu granicznego SE-LAW jako uzasadnienie przyjętych założeń i motywacja do dalszych prac w tym obszarze.

2. Zagadnienia inżynierii wymagań istotne dla prawników

Inżynieria wymagań powstała dla wsparcia działań informatyków podczas wytwarzania produktu na zamówienie. Niektóre dokumenty, w szczególności Specyfikacja Wymagań Systemowych, mogą być wykorzystywane jako załącznik do kontraktu stanowiący opis systemu, którego oczekuje klient, a dostawca zobowiązuje się go wykonać. Inżynieria wymagań jest działem inżynierii oprogramowania, który dotyczy pozyskiwania, analizy, specyfikacji i zarządzania zmianą wymagań [4].

Wymagania względem systemu (oprogramowania) opisują funkcje udostępniane przez system, wymaganą jakość systemu oraz ograniczenia w jakich system ma działać. Są one pozyskiwane, identyfikowane, analizowane i specyfikowane, ponieważ najczęściej nie jest oczywiste jaki system powinien być zbudowany.

Wyróżniane są następujące typy wymagań:

- Wymagania ogólne, sytuujące system w kontekście, np. jaki rodzaj systemu ma być zbudowany, jaka organizacja jest odbiorcą i jakie będą grupy użytkowników systemu, jaki rodzaj działalności system ma wspomagać.
- Funkcjonalne wymagania względem systemu – opisują, co system ma robić; przykładami mogą być funkcje, które system udostępnia użytkownikom; dane, które system ma przechowywać, obliczenia wykonywane przez system, wydruki i ich format;
- Poza-funkcjonalne (niefunkcjonalne) wymagania względem systemu - są związane z jakością systemu, np. wymaganie na wydajność określające jak szybko system ma działać, wymagania na ochronę danych związane z uwierzytelnieniem i ograniczeniem dostępu do zasobów; wymagania dotyczące ergonomii związane z łatwością użycia systemu;
- Wymagania eksploatacyjne – dotyczące środowiska działania systemu, np. system operacyjny, współpraca z innymi systemami;
- Wymagania projektowo-wdrożeniowe, np. wykonanie systemu według określonej metodyki, dokumentacja według określonego standardu, sposób oraz termin wdrożenia.

Znane są następujące problemy z wymaganiami:

- Problemy z określeniem wizji systemu, jego celu i zakresu;
- Rozproszona wiedza na temat wymagań - wiele źródeł informacji;
- Trudności pozyskania wymagań, np. trudność dostępu do przyszłych użytkowników systemu, niechęć użytkowników do współpracy, trudności ze wypowiedzeniem wymagań,
- Sprzeczne wymagania różnych grup użytkowników;
- Trudności w precyzyjnym sformułowaniu wymagań, aby można było jednoznacznie zinterpretować;
- Zmienność wymagań.

Stosowane są następujące techniki pozyskiwania wymagań:

- Studia literatury dotyczącej przedmiotu zawierającej podstawowe informacje na temat dziedziny problemowej;
- Identyfikacja udziałowców, czyli osób lub dokumentów mających pewien wpływ na powstający system lub punkt widzenia systemu;
- Wywiady i dyskusje z przyszłymi użytkownikami;
- Ankiety stosowane w celu pozyskania wymagań lub opinii użytkowników na temat powstającego systemu;
- Analiza dokumentów obowiązujących w danej organizacji;
- Analiza istniejącego systemu;
- Obserwacje przyszłych użytkowników systemu podczas ich pracy;
- Opis kontekstu działania systemu, np. warunki w jakich ma działać system;
- Symulacje punktów widzenia, stosowane w celu stwierdzenia, czy system będzie spełniał oczekiwania określonej grupy użytkowników;
- Prototypowanie – prezentacja użytkownikom 'atrapy' systemu, np. samego interfejsu użytkownika w celu pozyskania ich opinii lub dodatkowych wymagań;
- Prezentacje u dostawców podobnych systemów, stosowane w celu zapoznania się z podobnymi rozwiązaniami.

Pozyskiwanie wymagań jest powiązane z opisem organizacji, którego sformalizowaną formę stanowi modelowanie biznesowe.

Analiza wymagań ma na celu usunięcie nadmiarowości, sprzeczności i konfliktów oraz ocenę kompletności wymagań. Jest ona powiązana z analizą systemów i mają w niej zastosowanie metody modelowania systemu (więcej informacji w rozdziale 3).

Specyfikacja wymagań jest działalnością mającą na celu opis wymagań. Dokument Specyfikacji Wymagań Systemowych (SWS) jest oficjalną deklaracją tego, co jest wymagane od systemu. Może on służyć jako podstawa prac projektowych, podstawa ogłoszenia przetargu lub podstawa podpisania kontraktu. Jest on również punktem odniesienia podczas rozstrzygania spraw spornych. SWS zawiera informacje ogólne o systemie oraz dla każdego wymagania podaje: jego identyfikator, opis, źródło, priorytet, powiązania z innymi wymaganiami. Specyfikacja wymagań powinna być zweryfikowana i zatwierdzona przez klienta.

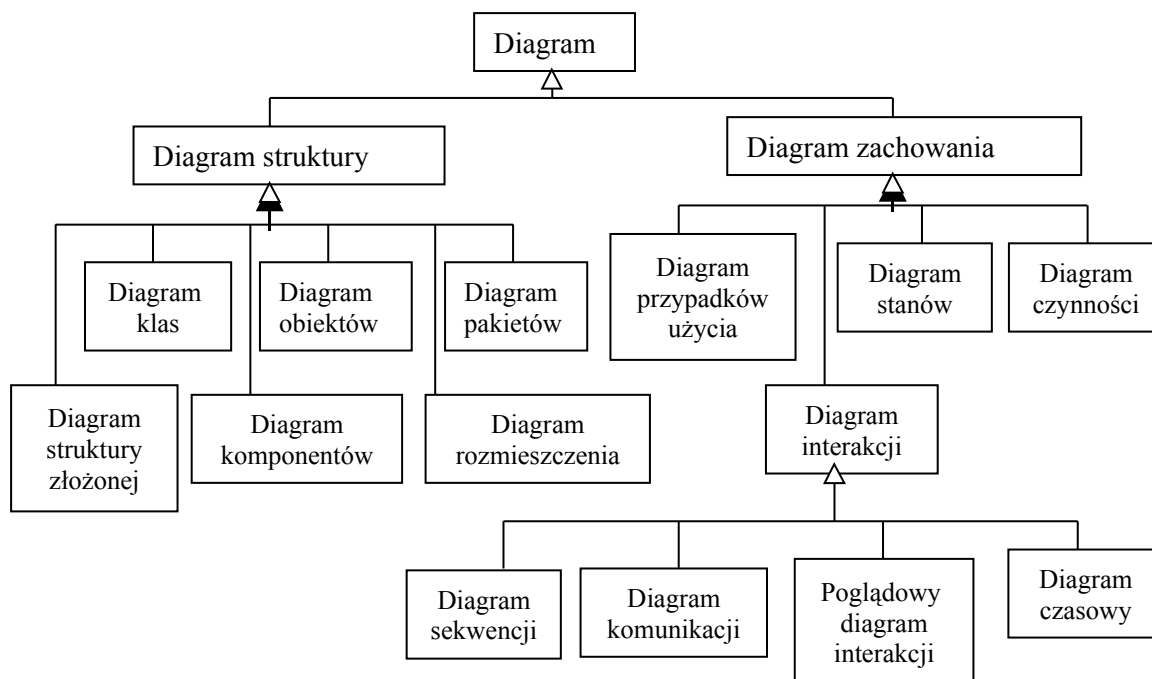
Zarządzanie zmianą wymagań dotyczy sytuacji, gdy po ustaleniu zakresu systemu i po rozpoczęciu prac związanych z jego wytwarzaniem wymagania ulegają zmianom. Przyczyną może być zmiana oczekiwań, zmiana technologii, zmiana celów, wykryte defekty itp. Wprowadzenie zmiany wymagania może mieć różnorakie konsekwencje organizacyjno-finansowe, wobec czego wymaga akceptacji obu stron. W praktyce decyzję podejmuje tzw. komitet zmian, a po akceptacji zmiany następuje procedura informowania zespołu o zmianie i podpisywane są aneksy do umowy.

Pozyskanie i specyfikacja wymagań jest złożonym zadaniem i wymaga odpowiednich nakładów. Jakość wymagań i stopień ich spełnienia zależy od współpracy z przedstawicielami klienta w trakcie pozyskiwania i weryfikacji wymagań, oceny prototypów oraz testowania wstępnych wersji systemu.

3. Modelowanie i analiza systemów

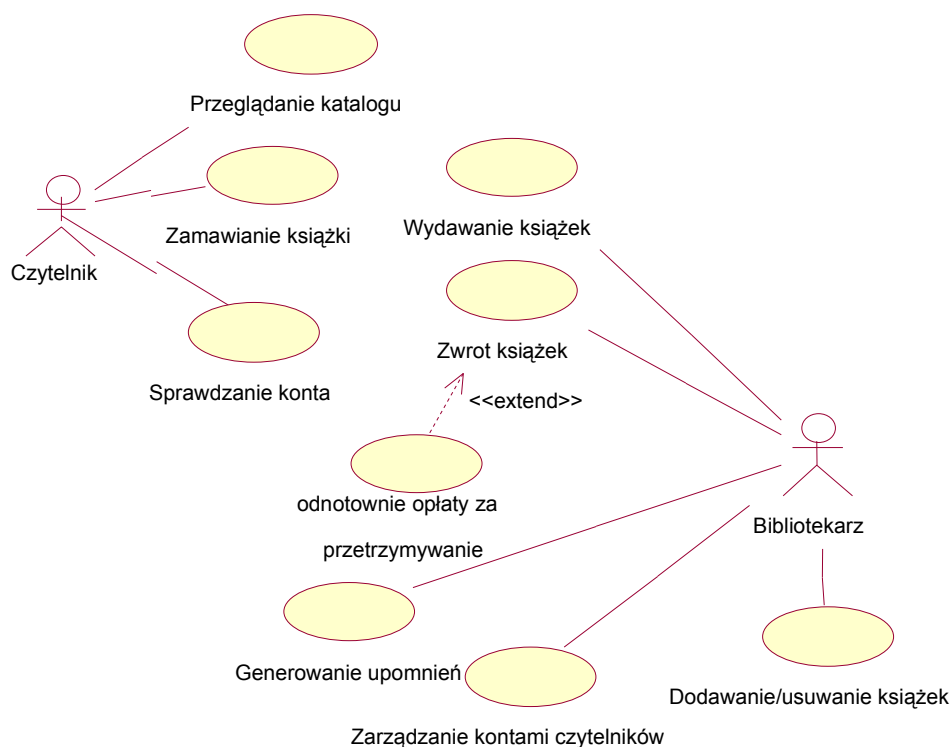
Model jest to reprezentacja rzeczywistości lub systemu z pewnego punktu widzenia i na określonym poziomie abstrakcji. Modelowanie ułatwia proces analizy i projektowania systemów poprzez wspomaganie wykonawców systemu w radzeniu sobie ze złożonością systemów oraz ukierunkowania ich myślenia na odpowiednie aspekty systemu. Przykładem zastosowania modelowania w fazie analizy systemów jest wspomaganie wykrywania problemów wśród dużej liczby wymagań, które mogą być sprzeczne, nakładające się, niekompletne lub nieprecyzyjnie sformułowane. Ponadto dzięki temu, że modele dostarczają precyzyjnego opisu systemu, są one dobrą podstawą prac weryfikacyjnych. Na podstawie modeli można przewidywać niektóre własności systemów. Dokumentacja z zastosowaniem modeli jest przydatna w komunikacji pomiędzy członkami zespołu projektowego, jak również z klientem. Dokumentacja ta jest przydatna także podczas utrzymania systemu, gdyż ułatwia osobom wprowadzającym zmiany zrozumienie wymagań oraz decyzji projektowych wykonawców oprogramowania.

Obecnie standardem w zakresie modelowania jest UML (ang. OMG Unified Modeling Language) [2]. Jest on językiem służącym do wizualnego przedstawiania określonych cech oprogramowania z wielu punktów widzenia. Na rys. 1 pokazano taksonomię diagramów UML 2.1. Zawiera ona typy diagramów, które umożliwiają opis zarówno struktury systemu, jak i jego zachowania; umożliwiają wykonanie zarówno poglądowego obrazu systemu, jak i specyfikację szczegółów.



Rys. 1. Taksonomia diagramów UML 2.1

Przykładem prostego, a zarazem często stosowanego diagramu jest diagram przypadków użycia. Pokazuje on użytkowników systemu (tzw. aktorów, symbol „ludzika”) oraz funkcje, które system dla nich udostępnia (tzw. przypadki użycia, symbol elipsy). Diagram przedstawiony na rys. 2 pokazuje funkcjonalność systemu wspomagającego wypożyczalnię książek. Czytelnik może za pomocą systemu przeglądać katalog, zamawiać książki i sprawdzać stan swojego konta czytelniczego, a bibliotekarz odnotowuje w systemie wydanie książki i jej zwrot (z ewent. odnotowaniem opłaty za przetrzymywanie książki), generuje upomnienia, zarządza kontami czytelników oraz dodaje i usuwa opisy książek z systemu.



Rys. 2. Diagram przypadków użycia dla systemu wspomagającego wypożyczalnię książek

Historie sukcesu pokazują przykłady zastosowania UML z sukcesem w wielu dużych projektach informatycznych na całym świecie oraz na ekonomiczną opłacalność tego typu inwestycji. Wśród korzyści wymieniane są: możliwość wykonania dużych projektów, skrócenie czasu dostarczenia rozwiązania, zmniejszenie kosztów oraz możliwość pracy w rozproszonych zespołach. Warto zauważyć, że sukces wymaga systematycznego i kompleksowego podejścia.

Wyniki badań ankietowych w polskich firmach informatycznych wskazują na następujące korzyści z zastosowania modelowania: poprawa dokumentacji produktu i dostępności specyfikacji, poprawa jakości oprogramowania, zwiększenie możliwości weryfikacji, zwiększenie możliwości przetwarzania specyfikacji, poprawa komunikacji w zespole oraz możliwość nadbudowania bardziej zaawansowanych technologii. Efektywność zastosowania zależy od zaawansowania technologii, stopnia dopasowania do specyfiki firmy oraz integracji z innymi narzędziami.

4. Wpływ wymagań i modeli analitycznych na jakość systemu

Brak specyfikacji wymagań względem systemu uniemożliwia weryfikację, czy system został wykonany zgodnie z określonymi wcześniej oczekiwaniami odbiorców. W sytuacji braku specyfikacji wymagań nie ma podstaw do szacowania pracochłonności jego wykonania, nie ma wytycznych dla wykonawców systemu dotyczących funkcji systemu, nie ma wytycznych dla testerów, jak sprawdzić jakość systemu. Niewyspecyfikowane wymagania jakościowe zazwyczaj nie są realizowane. Im wyższa jest jakość specyfikacji wymagań, tym mniejsze prawdopodobieństwo wystąpienia nieporozumień w jej interpretacji.

Jakie kryteria jakości powinny być spełnione dla Specyfikacji Wymagań Systemowych (SWS)? Wyróżniane są następujące atrybuty [3]:

- Jednoznaczność – SWS jest jednoznaczna, jeżeli wszystkie wymagania w niej zawarte są jednoznaczne, tzn. mają jedną interpretację semantyczną;
- Kompletność – SWS jest kompletna, jeżeli zawiera wszystkie wymagania klienta i każde wymaganie jest dokładnie opisane;
- Poprawność – SWS jest poprawna, jeżeli każde wymaganie w niej zawarte zostało przeanalizowane i potwierdzone przez klienta;
- Spójność – SWS jest spójna, jeżeli nie występują sprzeczności pomiędzy zawartymi w niej wymaganiami;
- Weryfikowalność – SWS jest weryfikowalna, gdy określone zostały procedury umożliwiające sprawdzenie, czy w wykonanym systemie wszystkie wymagania są spełnione;
- Modyfikowalność – SWS jest modyfikowalna, jeżeli jej struktura i styl są takie, że jakakolwiek zmiana w wymaganiach może być dokonana łatwo, kompletnie i spójnie;
- Śladowość – wymagania mają jasno określone źródło, wymagania są określone i istnieją powiązania pomiędzy różnymi wymaganiami w SWS i w innych dokumentach.

Badania projektów pokazują, że ponad połowa defektów występujących w oprogramowaniu wynika z niewłaściwego wyspecyfikowania wymagań. Dodatkowo, koszt naprawy takiego defektu wzrasta (według niektórych danych nawet dziesięciokrotnie) po przejściu do następnych faz procesu wytwarzania oprogramowania. W celu wyeliminowania tych defektów warto wykonać dokładną analizę systemu z zastosowaniem metod modelowania.

Jakość modeli można oceniać pod względem ich wyglądu, zawartości oraz dopasowania do odbiorcy dokumentacji. W zakresie wyglądu diagramów można oceniać: elegancję rozmieszczenia elementów na diagramie, zrozumiałość nazw, dokładność opisów poszczególnych elementów, odpowiedni poziom abstrakcji i brak nadmiarowości. Ta grupa atrybutów wpływa na czytelność dokumentacji lub problemy z jej zrozumieniem. W zakresie zawartości modeli można sprawdzać ich kompletność, poprawność, spójność, precyzję oraz zgodność z celami i wizją systemu. Ta grupa kryteriów jest bezpośrednio związana z defektami, które dotyczą wytwarzanego oprogramowania. W kwestii dopasowania do odbiorcy dokumentacji można oceniać, czy zawierają one wszystkie informacje, których ta osoba poszukuje oraz czy dany odbiorca jest w stanie zinterpretować tę dokumentację z zastosowaniem modeli.

5. Należyta staranność w zakresie wymagań i analizy systemów

Na ocenę, czy system informatyczny wykonano dokładając należytej staranności w zakresie wymagań i analizy systemu, wpływają następujące elementy:

- Zadania wykonywane przez osoby z odpowiednimi kompetencjami i predyspozycjami – zarówno pozyskiwaniem wymagań jak i analizą systemów powinny zajmować się osoby przeszkolone pod względem technik inżynierii wymagań oraz modelowania systemów, które dodatkowo posiadają predyspozycje do wykonywania takich zadań, np. komunikatywność, analityczne myślenie, zdolność do uporządkowania chaotycznie formułowanych myśli, precyzja, odporność na stres;
- Zadania wykonywane w odpowiednich warunkach projektowych (czas, budżet) – czasami w firmach informatycznych zadania analityczno-projektowe nie są doceniane z tego powodu, że nie powodują bezpośredniego przyrostu kodu programu, czego skutkiem jest marginalne potraktowanie ich w planach projektu – jednak skutkiem takiego podejścia jest słabe określenie celów i zakresu systemu oraz problemy z wymaganiami;
- Skuteczna komunikacja z przedstawicielami klienta w celu pozyskania i weryfikacji wymagań (co dotyczy zarówno strony dostawcy jak i strony klienta);
- Zastosowanie odpowiednich technik inżynierii wymagań - w zależności od sytuacji użyteczne lub wręcz możliwe do zastosowania są różne techniki pozyskiwania wymagań;
- Odpowiednia konfiguracja procesu analizy systemów - efektywne zastosowanie UML w firmie informatycznej wymaga zrozumienia roli modelowania w projekcie informatycznym oraz odpowiedniej konfiguracji działań, produktów, odpowiedzialności personelu i narzędzi wspomagających modelowanie;
- Wykonanie odpowiedniej dokumentacji (w szczególności Specyfikacji Wymagań Systemowych) i pozyskanie akceptacji klienta.

Pozyskiwanie i weryfikacja specyfikacji wymagań jest działalnością, która odbywa się we współpracy wykonawców oprogramowania z klientem. Bardzo ważne jest zaangażowanie klienta (a w szczególności przyszłych użytkowników systemu) w projekt. Czasami mówi się wręcz o osobnym projekcie odbywającym się w firmie klienta. Główne zadania związane z wymaganiami wykonuje wykonawca, ale klient powinien mu dostarczyć informacji na temat kontekstu działania systemu, np. struktury organizacyjnej, stosowanych procedur i wzorów dokumentów, wykonywanych zadań i związanych z nimi problemów oraz na temat swoich oczekiwań względem systemu. Znane są przypadki, że oprogramowanie nie mogło być wykonane w terminie z powodu niedostarczenia przez klienta opisu procedur, które miał wspomagać zamówiony system. Odpowiedzialność za inicjowanie pozyskiwania wymagań

oraz za profesjonalne wykonanie analizy ponosi wykonawca, natomiast klient ponosi odpowiedzialność za dostarczenie informacji oraz weryfikację przedstawionej dokumentacji.

6. Perspektywa dalszych prac: obiekt graniczny SE-LAW

W celu ujęcia zagadnień na pograniczu informatyki i prawa proponowane jest zastosowanie obiektu granicznego SE-LAW przez analogię do obiektów granicznych SE-HCI (ang. boundary object) służących integracji inżynierii oprogramowania i projektowania interfejsów użytkownika [1] (ang. software engineering - SE and human-computer interaction - HCI).

Zagadnienie integracji technik inżynierii oprogramowania i technik projektowania interfejsów użytkownika wynika z obserwacji braku spójności terminologicznej i braku dopasowania metodologicznego technik, które mają zastosowanie do tego samego 'bytu' – do oprogramowania i są stosowane w jednym przedsięwzięciu – w projekcie informatycznym. Wśród przyczyn tego zjawiska wymieniane są następujące fakty: techniki z obu grup były opracowywane przez niezwiązane ze sobą grupy badaczy, prezentowane były zazwyczaj na niezależnych konferencjach, wypracowały swoje własne, niezależne modele i terminologię, koncentrują się na innych aspektach jakości, wymagają innych predyspozycji i umiejętności, nauczane są na osobnych przedmiotach, a wyniki i metody wypracowane przez jedną z tych grup są niedoceniane przez drugą. Różnice na poziomie dyscyplin powodują problemy w praktycznych zastosowaniach, np. brak spójnej wizji systemu lub brak efektywności prac projektowych. W tej sytuacji zaproponowano pojęcie obiektu granicznego, jako pewną abstrakcję wspólnej przestrzeni pojęć, wspólnej wiedzy, wspólnych produktów i założeń, które mogą służyć do integracji i koordynacji działań osób lub zespołów posiadających różne specjalizacje. W praktyce znajdują zastosowanie trzy grupy obiektów granicznych: ludzie, np. kierownicy projektów, osoby przeszkolone w obu dyscyplinach; procesy, np. proces komunikacji i podejmowania decyzji, spotkania, 'burze mózgów'; oraz specyfikacje i dokumenty, np. prototypy interfejsów, specyfikacje, raporty.

Pomiędzy inżynierią oprogramowania a prawem występują jeszcze większe różnice w zakresie wiedzy, specjalistycznej terminologii, stosowanych podejść i metod, a także na poziomie sposobów myślenia osób reprezentujących te dwie profesje. Z drugiej strony, konieczne jest porozumienie i współpraca, aby skutecznie rozwiązywać problemy pojawiające się na przecięciu tych dwóch dyscyplin. Przykładem takich zagadnień są: podpisywanie umów dotyczących realizacji systemów informatycznych, rozstrzyganie o należytej staranności w przypadku konfliktów, tworzenie i udoskonalanie regulacji prawnych powiązanych z informatyką, czy też rozstrzygnięcia związane z tematyką przestępstw internetowych. Obiekt graniczny SE-LAW mógłby obejmować: elementy wiedzy o inżynierii oprogramowania dla prawników; elementy wiedzy o prawie dla informatyków; ustawy powiązane z systemami informatycznymi i ich wytwarzaniem, inne dokumenty, stan wiedzy w tym obszarze, przypadki orzeczeń itp. Tworzenie takiego obiektu granicznego ma prowadzić do powstania nowej specjalizacji, bez obaw o to, że prawnicy przejmą zadania informatyków, a informatycy będą się zajmowali rozstrzyganiem spraw sądowych. Jestem przekonana, że opracowanie i upowszechnienie tej wiedzy może się przyczynić do wzrostu staranności i profesjonalizmu podczas wytwarzania oprogramowania, do poprawy jakości systemów i do wzrostu satysfakcji użytkowników.

Literatura:

1. *Bridging the SE & HCI Communities*, <http://www.se-hci.org/bridging/>
2. *OMG Unified Modeling Language*, <http://www.uml.org>

3. Redmill F., *Inżynieria wymagań*, w: J. Górski (red.), Inżynieria oprogramowania w projekcie informatycznym, MIKOM, 1999.
4. Sommerville I., *Inżynieria oprogramowania*, WNT, 2003.



Anna Bobkowska

Anna Bobkowska jest adiunktem w Katedrze Inżynierii Oprogramowania Wydziału ETI Politechniki Gdańskiej.

Do swoich ciekawszych doświadczeń zawodowych zalicza:

- Pracę nad rozprawą doktorską pt. „Prognozowanie Jakości Oprogramowania na Podstawie. Modeli UML”
- Udział w pracach badawczych projektu „Generyczny model protokołów telesterowania” dla firmy Transmmit Oy (Finlandia)
- Udział w projektach UE Tempus
- Udział w badaniach grantu KBN pt. „Metodologia oceny jakości oprogramowania”
- Dwukrotny staż w University of Kent w ramach projektu SEGRAVIS
- Organizacja trzech sesji interdyscyplinarnej inżynierii oprogramowania na konferencjach „Technologie Informacyjne”.