

## **Tworzenie i wdrażanie produkcyjnego oprogramowania w środowisku akademickim**

Janina Mincer-Daszkiewicz  
*Instytut Informatyki, Uniwersytet Warszawski*

### **Streszczenie**

*Projekt Tempus NET i system USOS przedstawiono na KKIO'2001. USOS to ogólnouniwersytecki system do obsługi spraw studiów, realizowany na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego przez pracowników i studentów Wydziału w ścisłym sprzężeniu z dydaktyką — w ramach zajęć z inżynierii oprogramowania, baz danych i projektowania systemów informacyjnych oraz jako prace licencjackie i magisterskie.*

*Tematem tegorocznego artykułu są dwa zagadnienia związane z systemem USOS. Pierwsze dotyczy technik, metod i narzędzi inżynierii oprogramowania stosowanych podczas realizacji projektu. Główną metodyką jest CDM Fast Track, będący zalecanym przez Oracle standardem dla przedsięwzięć realizowanych w technologii RAD. Na etapie analizy wymagań i projektowania od niedawna stosuje się UML wspierany przez narzędzia Rational Rose. Z kolei projekt implementacji jest zapisywany przy użyciu narzędzi firmy Oracle, które służą również do szybkiego generowania prototypów. Sposób realizacji projektu jest w wielu punktach zgodny z praktykami wydajnego programowania.*

*Drugie zagadnienie wiąże się z przebiegiem wdrożenia systemu USOS w skali uczelni. System funkcjonujący już prawie dwa lata na Wydziale MIM jest stopniowo wdrażany na innych wydziałach Uniwersytetu Warszawskiego. W skład zespołu wdrożeniowego wchodzi osoby uczestniczące w rozwijaniu systemu oraz pracownicy Działu Aplikacji Komputerowych oraz Biura Spraw Studenckich UW.*

*Dalszy rozwój systemu USOS będzie się odbywał pod patronatem Uniwersyteckiego Centrum ds. Informatyzacji, zrzeszającego 15 polskich uniwersytetów. Zainteresowanie systemem i chęć jego wdrożenia wyraziła także Politechnika Warszawska.*

## 1. Wprowadzenie

Uniwersytecki System Obsługi Studiów (USOS) powstaje na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego (MIM UW). Prace nad systemem rozpoczęto w 1999 roku w ramach projektu Tempus NET, w którym uczestniczyły wszystkie polskie uniwersytety. Szczegóły projektu, jego cele, zadania i tryb realizacji (w ujęciu ogólnym) przedstawiono na KKIO'2001([5]).

W chwili obecnej, czyli pod koniec roku akademickiego 2001/02 USOS jest używany do obsługi spraw studiów na trzech wydziałach trzech polskich uczelni. Na kilku innych uczelniach rozpoczęto proces wdrożenia z założeniem, że system zacznie działać produkcyjnie z początkiem roku akademickiego 2002/03. Na Uniwersytecie Warszawskim są prowadzone intensywne prace nad wdrożeniem systemu w skali uczelni.

USOS jest realizowany przez pracowników i studentów Wydziału MIM UW w ścisłym sprzężeniu z dydaktyką — w ramach zajęć z inżynierii oprogramowania, baz danych i projektowania systemów informacyjnych oraz jako prace licencjackie i magisterskie ([8]). Głównym celem przedsięwzięcia jest oczywiście wytworzenie oprogramowania, bez którego trudno sobie obecnie wyobrazić efektywną pracę dziekanatów wyższych uczelni w dobie rozwiniętej indywidualizacji studiów i wzmożonej mobilności studentów. Nie mniej ważny, zważywszy na miejsce prowadzenia projektu, jest cel dydaktyczny. Trafiła się niepowtarzalna okazja wciągnięcia studentów do prac w rzeczywistym projekcie, ale też ten udział powinien się wiązać z poznawaniem przez nich metod i narzędzi inżynierii oprogramowania oraz przyswajaniem **dobrych praktyk**.

Przedmiotem pracy są dwa zagadnienia związane z systemem USOS. Pierwsze dotyczy technik, metod i narzędzi inżynierii oprogramowania stosowanych podczas realizacji projektu. Główną metodyką jest CDM Fast Track ([3]), będący zalecanym przez Oracle standardem dla przedsięwzięć realizowanych w technologii RAD. Dzięki zakupowi Rational Rose, na etapie analizy wymagań i projektowania można stosować UML. Z kolei projekt implementacji jest zapisywany przy użyciu narzędzi firmy Oracle. Sposób realizacji projektu jest w wielu punktach zgodny z praktykami wydajnego programowania (ang. *extreme programming*).

Drugie zagadnienie wiąże się z przebiegiem wdrożenia systemu USOS w skali uczelni. System funkcjonujący już prawie dwa lata na Wydziale MIM jest stopniowo wdrażany na innych wydziałach Uniwersytetu Warszawskiego. W skład zespołu wdrożeniowego wchodzi osoby uczestniczące w rozwijaniu systemu oraz pracownicy centralnej administracji uniwersyteckiej. Wdrożenie wiąże się z koniecznością opracowania wspólnych słowników, uzgodnieniem zawartości istniejącej na UW bazy finansowo-księgowej z bazą USOS, konwersją danych związanych z tokiem studiów ze starych baz wydziałowych, ujednoczeniem procedur. Sprawą niezmiernie ważną jest opracowanie i wdrożenie właściwej polityki bezpieczeństwa, a co za tym idzie zdefiniowanie uprawnień użytkowników systemu. Wreszcie konieczne jest przygotowanie spersonalizowanych zestawów filtrów pozwalających na wygodne poruszanie się użytkowników po wspólnym repozytorium danych.

W rozdziale 2 pracy przedstawiamy metody i narzędzia inżynierii oprogramowania stosowane podczas realizacji projektu. Tematem rozdziału trzeciego jest przebieg wdrożenia systemu na Uniwersytecie Warszawskim. Rozdział czwarty zawiera podsumowanie.

## 2. USOS a inżynieria oprogramowania

### 2.1 Metodyka CDM Fast Track

W standardowym podejściu do projektowania systemów, któremu odpowiada tzw. model kaskadowy (por. [6]), etapy projektu (analiza wymagań i specyfikacja systemu, projektowanie, implementacja, testowanie, wdrożenie, eksploatacja) następują po sobie w ściśle określonym porządku liniowym. W rzeczywistym świecie zachowanie takiego porządku nie jest jednak najczęściej możliwe. Wadą modelu kaskadowego jest to, że nie uwzględnia on możliwości zmiany wymagań w trakcie trwania projektu.

Główną metodyką zastosowaną w projekcie USOS jest CDM Fast Track ([3]), będący zalecanym przez Oracle standardem dla przedsięwzięć realizowanych w technologii RAD. Metodyka ta jest oparta na DSDM, której cechą charakterystyczną jest dopuszczanie iteracyjnego powtarzania faz. W metodyce CDM Fast Track stosuje się następujące techniki:

- określanie ram czasowych (ang. *timeboxes*) i priorytetów,
- spotkania z użytkownikami (ang. *workshops*),
- tworzenie prototypów,
- iteracyjne dopracowywanie modułów systemu,
- dostarczanie częściowych rozwiązań,
- podział prac,
- udział w pracach projektowych użytkowników mogących samodzielnie podejmować ważne decyzje w dziedzinie tworzonego systemu.

W dalszej części rozdziału zostanie pokrótce omówiony sposób realizacji tych technik w projekcie USOS (bardziej szczegółowy opis można znaleźć w pracy magisterskiej [4]).

#### 2.1.1 Określenie ram czasowych i priorytetów

Techniki te stosuje się na każdym etapie przedsięwzięcia w omawianej metodyce. Pierwsza z nich pozwala na określenie punktu startowego i końcowego dla pewnego procesu, którego efektem musi być osiągnięcie jakiegoś konkretnego celu. Inaczej niż w modelu kaskadowym, efektem zdefiniowania pewnego okresu nie jest np. wykonanie jakiegoś modułu zajmującego się przetwarzaniem danych niewidocznym dla użytkownika. Tutaj musi to być całościowa, dostarczana użytkownikowi część aplikacji.

Dla każdego zdefiniowanego przedziału czasu i zadania należy określić listę priorytetów, którą powinno się w tym czasie i zadaniu zrealizować. Najważniejsze są oczywiście

wymagania kluczowe dla działania systemu, które muszą być zrealizowane, aby wdrażanie systemu miało w ogóle sens. Ponieważ życie uczelni obfituje w przeróżne wydarzenia mające wpływ na przebieg studiów, a USOS od początku był wdrażany na wydziale MIM UW, ramy czasowe dla wielu modułów systemu wyznaczane są częstokroć przez wspomniane wydarzenia. Przykładem może być konieczność zakończenia prac nad modulem do liczenia średnich przed rozpoczęciem rejestracji na seminaRIA magisterskie. Przy projektowaniu systemu zespół wykonawczy ściśle współpracuje z użytkownikami, owocem tej kooperacji jest więc bardzo dobre określenie priorytetów wymagań i w przypadku, gdy istnieje niebezpieczeństwo nieukończenia zadania w terminie, przesuwanie mniej istotnych wymagań na później.

### **2.1.2 Spotkania z użytkownikami**

Spotkania z użytkownikami stanowią efektywną technikę pomocną w zbieraniu informacji od osób mających wykorzystywać system i w podejmowaniu decyzji projektowych. Pozwalają na uniknięcie nieporozumień na styku użytkownik-projektant. Owocem takich spotkań są ustalenia, które pozwalają zminimalizować ryzyko niepowodzenia jakiejś części przedsięwzięcia.

W czasie pracy nad USOS w ramach projektu Tempus organizowano cotygodniowe spotkania, w których uczestniczyło kierownictwo przedsięwzięcia, projektanci oraz zaproszeni użytkownicy. Ponadto, jeśli tylko zachodziła taka potrzeba, organizowano spotkania dodatkowe w celu zaprezentowania użytkownikom postępów prac nad jakimś modulem oraz zebrania ich uwag i komentarzy. Bardzo ważną rolę odgrywały konferencje robocze, grupujące przedstawicieli uczelni uczestniczących w projekcie. Takie konferencje odbywały się co kilka miesięcy, umożliwiając przekazywanie uczestnikom informacji o postępach projektu, weryfikowanie realizowanych w systemie wymagań funkcjonalnych oraz ustalanie długoterminowych kierunków dalszych prac.

### **2.1.3 Tworzenie prototypów**

Prototypowanie jest sprawdzoną techniką komunikacji z użytkownikami. Prototypy powinny być dostarczane użytkownikom w krótkim czasie po określeniu przez nich pierwszych wymagań funkcjonalnych. Umożliwia to łatwe wprowadzanie zmian, w przypadku gdy potrzeby użytkownika nie zostały w pełni zrozumiane przez analityków.

Charakter projektu umożliwia szybkie i częste dostarczanie prototypów. Przekazuje się je użytkownikom, którzy dostarczają informacje zwrotne na temat poprawności, wygody używania, czytelności i intuicyjności interfejsu, brakujących funkcji pomocniczych, czy raportów. Prototypy zazwyczaj są pozbawione pewnych funkcji automatyzujących wykonywane czynności (np. można rozliczać studentów z programu indywidualnie, a brakuje wsparcia do rozliczania całych roczników), pomocniczych filtrów (do oglądania danych w różnych przekrojach: studentów wskazanego kierunku, czy etapu studiów), narzędzi do wyliczania globalnych statystyk. Taki tryb postępowania jest możliwy jedynie dzięki temu, że użytkownicy systemu, decydenci, projektanci i programiści obracają się w tym samym środowisku uczelni wyższej. Jest to zarazem ten aspekt projektu, który chyba w największym stopniu decyduje o jego powodzeniu.

### **2.1.4 Iteracyjne dopracowywanie modułów**

Metodyka Oracle zakłada trzy iteracje każdej fazy przedsięwzięcia. Zadaniem zespołu w czasie pierwszej iteracji jest określenie i analiza potrzeb użytkowników, tj. zebranie możliwie jak największej liczby faktów koniecznych do zbudowania podstaw aplikacji, w drugiej iteracji zbiera się uwagi użytkowników wynikające z zapoznania się z wynikami pierwszej części i dodaje nowe wymagania, W tej fazie należy implementować skomplikowane reguły biznesowe. Iteracja trzecia to etap konsolidacji i wprowadzania zmian wymaganych przez użytkowników po drugiej iteracji.

W projekcie USOS motorem kolejnych iteracji są uwagi przekazywane przez użytkowników prototypów zgłaszających pomysły i propozycje modyfikacji. Wymagania nie są ostatecznie określone w pierwszej fazie projektu, lecz doprecyzowuje się je w trakcie kolejnych iteracji.

### **2.1.5 Dostarczanie częściowych rozwiązań**

Tę technikę stosuje się, aby skrócić czas oczekiwania użytkowników na kolejne moduły. W pracach projektowych oznacza to przede wszystkim, że funkcjonalność mniej istotna dla działania modułu może być implementowana w terminie późniejszym. Przykładowo, elementy modułu liczenia średnich realizowano w następującej kolejności:

- bazowa funkcja do liczenia średniej ocen ze wskazanego zbioru przedmiotów, dostępna jako procedura w SQL Plus;
- formularze stanowiące interfejs do wywoływania funkcji liczącej średnią dla wskazanego studenta lub grupy studentów;
- funkcje wspomagające oraz realizujące bardziej złożoną logikę, np. branie pod uwagę równoważności pewnych przedmiotów.

### **2.1.6 Podział prac**

Tę technikę stosuje się w celu podzielenia dużych przedsięwzięć na mniejsze części, które mogą być projektowane i implementowane w krótszych przedziałach czasowych oraz przez mniejsze grupy lub pojedynczych wykonawców. USOS daje się w naturalny sposób dzielić na podsystemy, które mają te cechy.

### **2.1.6 Udział w pracach projektowych użytkowników mogących samodzielnie podejmować ważne decyzje w dziedzinie tworzonego systemu**

Założenie to daje możliwość szybkiego i elastycznego reagowania na zmiany wymagań oraz zapewnia możliwość mobilizowania użytkowników do pewnych działań zwiększających szansę powodzenia przedsięwzięcia.

W USOS kierujący przedsięwzięciem są także jego użytkownikami i mają duże możliwości decyzyjne. Szefem projektu jest prodziekan Wydziału MIM, która może ustalać cele, określać priorytety wymagań, podejmować decyzje i egzekwować ich wykonanie. Będąc równocześnie nauczycielem akademickim w zakresie informatyki, może wziąć na siebie odpowiedzialność za dydaktyczny aspekt projektu.

## 2.2 Narzędzia

Metodyka CDM Fast Track, jak każda metodyka określająca zasady prowadzenia przedsięwzięcia RAD, musi mieć zapewnione wsparcie narzędziowe. Ponieważ metodyka ta jest standardem firmy Oracle, naturalne wydaje się stosowanie narzędzi tej właśnie firmy.

### 2.1.8 Narzędzia bazodanowe

Dane w USOS przechowywane są w systemie bazy danych Oracle8. Proces tworzenia tabel w tej bazie jest następujący:

- tworzony jest model związków encji w Oracle Designer,
- Oracle Designer generuje w swoim repozytorium wstępną postać tabel, sekwencji, indeksów itp.,
- wstępny projekt bazy danych jest podstawą dalszych prac projektowych; powstaje ostateczny projekt bazy danych, generowane są skrypty DDL,
- skrypty uruchamia się w produkcyjnej bazie danych.

Zmiany w wymaganiach mogą następować w dowolnej fazie przedsięwzięcia, mogą również wymagać przepisywania danych między tabelami. Konieczna jest zatem możliwość wprowadzania zmian w bazie nie tylko poprzez repozytorium, a poprzez ręcznie tworzone skrypty. Do testowania takich skryptów lub zapytań do bazy danych używamy darmowej wersji narzędzia TOAD (por. [9]).

### 2.1.9 Tworzenie formularzy i raportów

W procesie tworzenia formularzy i raportów możemy wyróżnić następujące etapy:

- tworzenie projektu modułu (formularz, raport) w Oracle Designer,
- generowanie gotowego modułu przez Oracle Designer przy użyciu odpowiednich plików zawierających opis standardu (por. [2]),
- dopracowanie modułu w narzędziach Oracle Forms lub Oracle Reports.

Opisany proces pozwala szybko wygenerować prototyp wymagany w stosowanej metodyce. Za pomocą Oracle Designera można w krótkim czasie zaprojektować zarówno formularz, jak i raport, który we wstępnej postaci można zaprezentować przyszłym użytkownikom. Dzięki ich uwagom można modyfikować prototyp już we wczesnej fazie jego powstawania, a więc nie wymaga to głęboko sięgających zmian.

## 2.3 Rational Rose i UML

Wsparcie narzędziowe istotnie ułatwia stosowanie metodyki w procesie konstrukcji oprogramowania. Narzędzia firmy Oracle pełnią w projekcie USOS kluczową rolę na etapie projektowania i implementacji modułów. Nie dają jednak wystarczającego wsparcia na wcześniejszym etapie analizy wymagań. Takie wsparcie zapewnia zakupiony

z funduszy Tempusa pakiet Rational Rose. Jest to zestaw narzędzi do modelowania aplikacji na wszystkich etapach jej rozwoju, wizualizacji elementów projektu, definiowania zachowania aplikacji, dokumentowania decyzji projektowych i realizacji wielu innych czynności w procesie tworzenia aplikacji. W Rational Rose językiem modelowania jest UML (ang. *Unified Modeling Language*). UML umożliwia:

- określenie zakresu systemu i jego głównych funkcji poprzez wskazywanie *aktorów* (użytkowników systemu) i opisywanie *przypadków użycia*,
- ilustrowanie realizacji przypadków użycia za pomocą *diagramów interakcji*,
- reprezentowanie statycznej struktury systemu przy użyciu *diagramów klas*,
- modelowanie zachowania obiektów za pomocą *diagramów przejść międzystanowych*,
- wyrażanie fizycznej architektury implementacji za pomocą *diagramów komponentów* i *diagramów wdrożenia*,
- rozszerzanie funkcjonalności przy użyciu *stereotypów*.

Ważne z punktu widzenia spójności stosowanych w projekcie metodyk i narzędzi jest to, że UML i Rational Rose, podobnie jak CDM Fast Track i Oracle, wspiera podejście obiektowe i jest zgodne z DSDM, czyli dopuszcza iteracyjny powtarzanie faz.

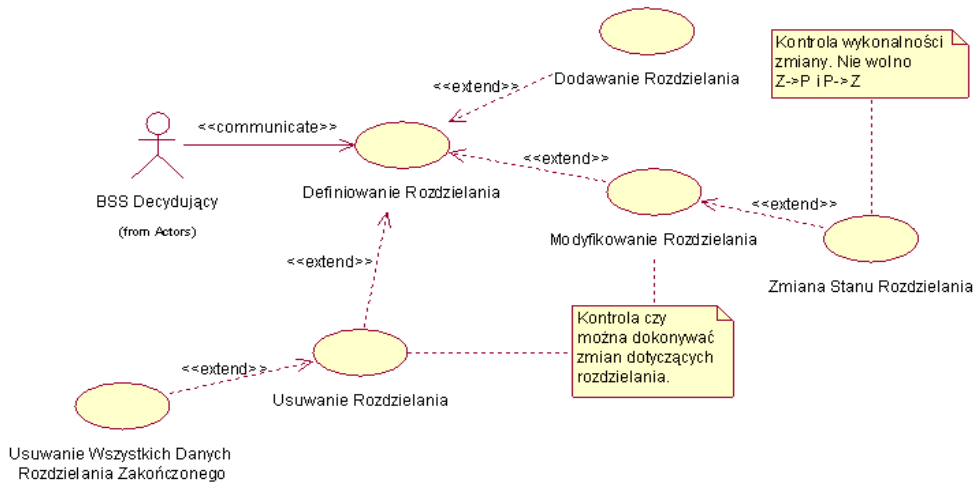
Pierwszym modułem USOS, przy projektowaniu którego użyto UML i Rational Rose, jest moduł **Akademiki** (por. [7]). Jest on dopiero w trakcie realizacji, niemniej jednak został już częściowo zamodelowany w UML oraz wdrożono pierwszą prototypową część modułu. Jest ona przeznaczona dla Biura Spraw Studenckich (BSS) i służy do wprowadzania do bazy danych informacji o akademikach i oferowanych w nich miejscach, a także rozdzielania tych miejsc pomiędzy wydziały. Model tej części obejmuje wiele diagramów, przy czym do najważniejszych należą:

- **przypadki użycia**: jest ich bardzo wiele; przykładowy przypadek użycia pokazano na rys. 1 (*BSS – Podejmowanie decyzji – Definiowanie Rozdzielenia*).
- **przebieg czynności**: pełny diagram przepływu czynności dla modułu **Akademiki** pokazano na rys. 2.
- **kolejność wprowadzania danych**: przykładowy diagram definiujący kolejność wprowadzania danych słownikowych zawiera rys. 3.

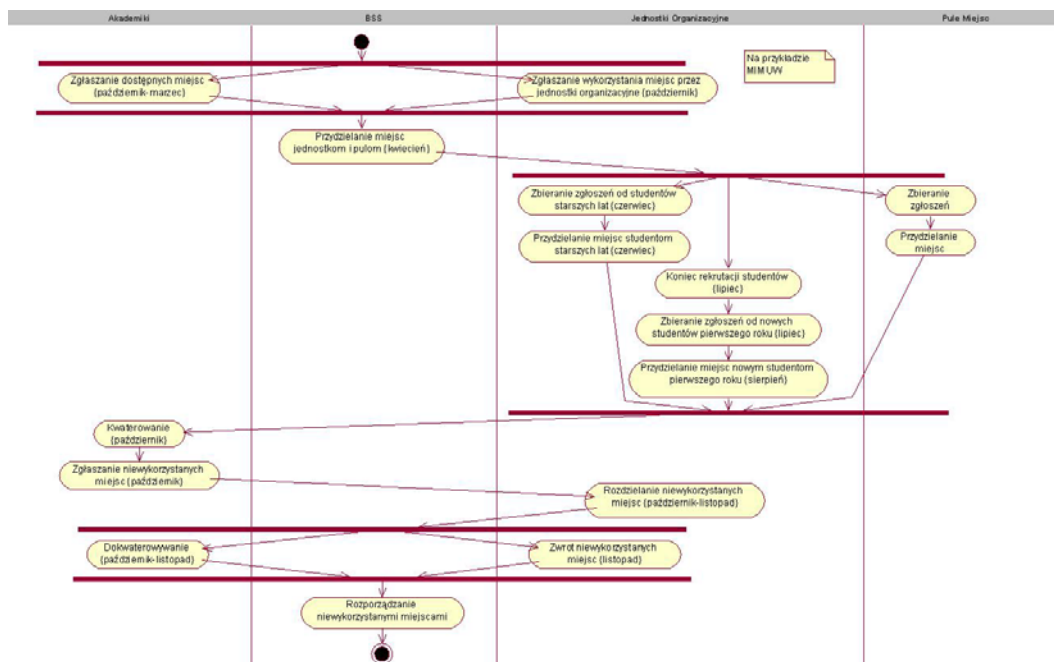
W trakcie powstawania są diagramy komponentów i diagramy realizacji przypadków użycia. Jak widać z załączonych przykładów, diagramy powstające na etapie analizy wymagań są, wyrażone w języku zrozumiałym dla użytkownika, ułatwiają myślenie o problemie i jego możliwych rozwiązaniach. Mogą więc pełnić bardzo użyteczną rolę wspomagając wymianę myśli między użytkownikami systemu, specjalistami do spraw studiów i projektantami. W przypadku projektu USOS ta interakcja jest tym prostsza, że wszystkie osoby zaangażowane w proces inżynierii wymagań obracają się w tym samym środowisku. Dzięki temu użytkownik może wpływać na końcowy produkt na bardzo wczesnym etapie projektu. Pozwala to uniknąć niepotrzebnych cykli, obejmujących oprócz fazy analizy wymagań również fazę implementacji. Na dodatek automatycznie uzyskuje się w ten sposób dokumentację projektu, także tę jej część, która może potem zostać rozwinięta w podręcznik użytkownika.

Wszystkie decyzje dotyczące rozdzielania miejsc w akademikach odnoszą się do konkretnych rozdziałów. Każde rozdzielanie odpowiada jednemu cyklowi dydaktycznemu i jest jego odzwierciedleniem na sprawy związane z akademikami. Fazę w jakiej znajdują się prace związane z danym rozdzielaniem określa jego stan (stan można modyfikować zawsze, ale tylko w określony sposób):

- Planowane (faza planowania, można wykonywać wszystkie sensowne modyfikacje)
- Aktywne (faza wykorzystywania, nie można dokonywać żadnych zmian)
- Zakończony (faza archiwizacji, nie można dokonywać żadnych zmian poza skasowaniem wszystkich danych)



Rysunek 1: Przypadek użycia BSS – Podejmowanie decyzji – Definiowanie Rozdzielania

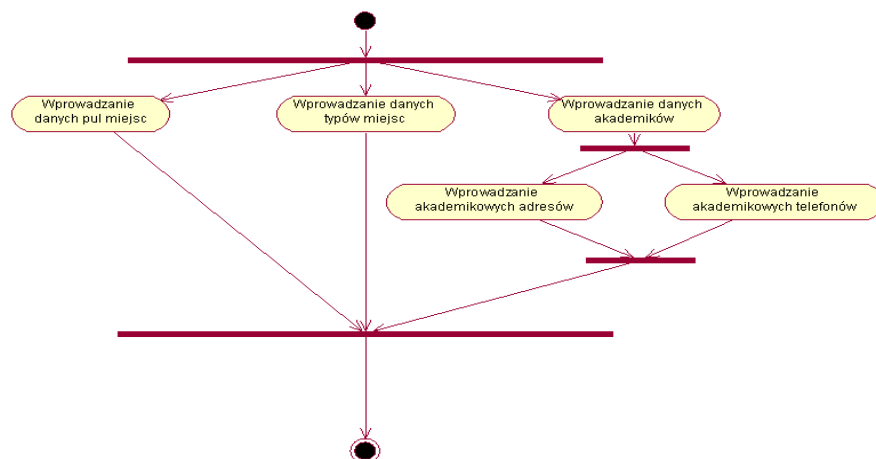


Rysunek 2: Przepływ czynności w module **Akademiki** – Rozdzielanie miejsc w akademikach



UML może być również stosowany do modelowania dalszych faz projektu, np. logicznej struktury bazy danych. Strukturę tę można również wygodnie opisać przy użyciu Oracle Designer i zasadniczo diagramy uzyskiwane z obu narzędzi są równoważne, nie widać więc na razie argumentów za zastąpieniem narzędzi Oracle na tym etapie.

Pewne rozszerzenia UML mogą być wykorzystane do modelowania internetowych modułów USOS. Jest kilka takich modułów: rejestracja na przedmioty, elektroniczny indeks, elektroniczne protokoły, ewaluacja pracowników dydaktycznych, rejestracja do grup zajęciowych, giełda miejsc w akademikach, weryfikacja dokumentacji składanej w celu uzyskania pomocy socjalnej.



Rysunek 3: Kolejność wprowadzania danych słownikowych

## 2.4 Wydajne programowanie

Praktyki stosowane w projekcie USOS są w wielu punktach zbieżne z zasadami **wydajnego programowania** (ang. *extreme programming*). Spośród zasad wydajnego programowania stosowanych w projekcie można wymienić następujące:

- **Pracuj ze swoimi klientami** – użytkownicy systemu (pracownicy administracji wydziału, nauczyciele akademicy, studenci) wchodzi w skład zespołu projektowego i pełnią aktywną rolę w określaniu wymagań i sposobu działania aplikacji, a także jej testowaniu i wdrażaniu.
- **Planuj** – planowanie jest obecne zarówno w zakresie długofalowym, jak i krótkofalowym, przy czym planowanie długofalowe jest z założenia ograniczane do zakresu ogólnych ram i kierunku rozwoju systemu, a projektowanie krótkofalowe jest dużo bardziej precyzyjne i dotyczy zadań realizowanych w skali 1–3 miesięcy.
- **Spotykaj się na krótko** – wszystkie spotkania organizowane w ramach projektu mają dobrze zdefiniowany grafik i ściśle określone ramy czasowe. W początkowej

fazie projektu projektanci i programiści spotykali się regularnie co tydzień, w terminie, który uniemożliwiał przedłużanie spotkań.

- **Stawiaj na prostotę** – w projekcie USOS jednym z ważniejszych celów projektowych jest dążenie do zapewnienia prostoty systemu, przejrzystości jego logiki, intuicyjności interfejsu. Nie zawsze jest to łatwe do zrealizowania, gdyż kłóci się z koniecznością zapewnienia jak największej elastyczności (należy pamiętać, że USOS ma służyć do obsługi spraw studiów na wielu wydziałach i uczelniach). Jesteśmy jednak przekonani, że nie można zapewnić elastyczności kosztem prostoty. Przekonanie takie wynika ze znajomości realiów środowiska uniwersyteckiego i świadomości, iż z aplikacją często będą pracować użytkownicy, których doświadczenie z zakresu obsługi komputera ogranicza się do niezbyt sprawnego poruszania się w środowisku Windows i Microsoft Office. Zdarzało się, że prototyp modułu uznawano za zbyt skomplikowany i zdecydowano się na opracowanie jego prostszej wersji.
- **Koduj w zgodzie ze standardami** – jedna z prac magisterskich napisanych w ramach projektu jest w całości poświęcona obowiązującym w projekcie standardom ([2]). Każdy nowy programista jest obowiązany zapoznać się z tymi standardami i stosować je w swoim kodzie.
- **Niech właścicielami będą wszyscy** – każdy uczestnik projektu może, w miarę potrzeby, dokonywać zmian w dowolnym module systemu. Ochronę dostępu zapewnia system kontroli wersji CVS (dla kodu części internetowej projektu) i system rezerwacji formularzy i raportów (dla części Oracle), przy czym integrowanie kodu odbywa się bardzo często.
- **Integruj stale** – integracja nowych wersji kodu z kodem produkcyjnym odbywa się praktycznie na bieżąco.
- **Często dostarczaj nowe dystrybucje** – nowe wersje oprogramowania są dostarczane użytkownikom często i odpowiadają niewielkim przyrostom funkcjonalności.

Techniki wydajnego programowania, których nie udało się zrealizować w projekcie to:

- **Koduj w parach** – studenci realizują w ramach projektu swoje prace magisterskie, więc zasadniczo nad jednym modulem pracuje jeden student. Ponieważ jednak studenci mają do swojej dyspozycji osobne laboratorium komputerowe, niejednokrotnie w ramach sąsiedzkiej pomocy wspólnie rozwiązują pojawiające się problemy programistyczne.
- **Zaczynaj od pisania testów** – tej praktyki nie udało się wdrożyć, m.in. ze względu na charakter aplikacji, która głównie opiera się na interakcji z użytkownikiem siedzącym przy terminalu. Jeden z pierwszych projektów mających na celu opracowanie narzędzi wspomagających automatyczne testowanie zakończył się niepowodzeniem (planowana praca magisterska jeszcze nie powstała).

Należy też podkreślić, że w pewnej sprzeczności z jedną z zasad wydajnego programowania, w projekcie USOS przykładamy dużą wagę do dokumentowania. Diagramy UML nie są zatem wyrzucane po zakończeniu fazy projektowej, lecz służą jako trwałe elementy dokumentacji. W skład dokumentacji projektu wchodzi diagramy

związków encji oraz schemat bazy danych utrzymywany w Oracle Designer, pisane przez studentów prace licencjackie i magisterskie, powstająca w miarę na bieżąco kilkupoziomowa dokumentacja użytkownika. Oznacza to oczywiście konieczność ich aktualizowania. Wysiłek ten nie idzie naszym zdaniem na marne. Zarówno skala projektu (oprogramowanie dla rozproszonych geograficznie dużych odbiorców), jak i charakter zespołu projektowo-programistycznego (studenci, którzy z naturalnych powodów spędzają w projekcie nie więcej niż dwa lata) wymagają istnienia dokumentacji.

### **3. Wdrożenie USOS w skali uczelni**

#### **3.1 Wprowadzenie**

USOS był przez ponad półtora roku używany do obsługi spraw studiów na Wydziale MIM, gdy rektor UW podjął decyzję, że system ma zostać wdrożony na całej uczelni. Trudności z wdrożeniem dużego systemu komputerowego w przedsiębiorstwie tej wielkości i o takim profilu działalności są wielorakie:

- jednostki organizacyjne UW są bardzo różnej wielkości, różnią się również znacznie w zakresie organizacji studiów;
- stopień komputeryzacji jednostek jest bardzo różny: niektóre łączą się ze światem zewnętrznym łąkami światłowodowymi, a inne za pomocą modemów; pracownicy administracji w większości mają dostęp do w miarę nowoczesnego sprzętu, ale tylko na niewielu wydziałach jest zapewniony masowy dostęp do komputerów pracowników dydaktycznych i studentów;
- na niektórych wydziałach do tej pory nie używano żadnego oprogramowania do obsługi spraw studiów, żadne dane nie są więc dostępne w postaci elektronicznej. Inne wydziały korzystały z lokalnych baz danych i wspierających je aplikacji.
- umiejętność posługiwania się komputerami wśród pracowników administracji jest najczęściej niezadowalająca. Nauka posługiwania się nową aplikacją, nawet działającą w popularnym w administracji środowisku Windows, niektórym przychodzi z dużym trudem;
- wydziały i instytuty cieszą się dużą autonomią, co czasami utrudnia koordynację działań w skali ogólnouczelnianej;
- wdrożenie wymaga dużego wysiłku, nie tylko samego zespołu wdrożeniowego, lecz również pracowników administracji wydziałowej i władz dziekańskich wydziału. System, żeby móc przynosić jakiegokolwiek korzyści, musi zostać skonfigurowany do potrzeb wydziału i wypełniony danymi. W etapie przejściowym niekiedy trzeba utrzymywać zarówno stare bazy danych, jak i nową. Decyzja o zaniechaniu wprowadzania danych do starych baz powoduje, że pewnych zadań nie daje się realizować (np. policzenia średniej, gdy część ocen jest w starej bazie, a część w nowej). W początkowej fazie system więcej wymaga od użytkowników niż daje w zamian.

- również w dalszych fazach wdrożenia może się okazać, że z punktu widzenia niektórych pracowników pracy nie ubywa, lecz wręcz przeciwnie. Czynności, które do tej pory były realizowane w pewnych działach są przenoszone w miejsce, gdzie jest możliwy dostęp do systemu; dane, które dotychczas funkcjonowały w postaci niestrukturalnych dokumentów edytora tekstów (np. plany zajęć) muszą zostać wprowadzone do systemu w sposób zgodny z jego logiką. Daje to docelowo dużo większe możliwości (np. wyszukiwania wolnych sal przez przeglądarkę WWW), ale z punktu widzenia konkretnego pracownika może być postrzegane jako zwiększenie obciążenia.
- różne są priorytety wydziałów: niektóre są zainteresowane głównie modulem do obsługi studiów płatnych, inne rejestracją studentów na zajęcia przez Internet, a jeszcze inne generowaniem raportów i statystyk dla GUS. Te czasami sprzeczne oczekiwania trzeba umieć pogodzić, bo raz zniechęcony klient nieprędko ponownie podejmie wysiłek wdrożenia. Entuzjazm i zaangażowanie ludzi jest zaś podstawą sukcesu każdego wdrożenia.

Bardzo ważne znaczenie dla powodzenia wdrożenia miała jasna i stanowcza deklaracja rektora UW, że USOS ma pełnić rolę systemu ogólnouczelnianego, wdrożenie w części ogólnej będzie finansowane z budżetu rektora i żadne lokalne aplikacje nie będą już wspierane ani rozwijane przez centralną administrację uniwersytecką. (np. lokalne bazy danych). Ta decyzja rektora ułatwiła również kontakty zespołu wdrożeniowego z pracownikami administracji centralnej.

### 3.2 Skład zespołu wdrożeniowego

W skład zespołu wdrożeniowego wchodziły osoby uczestniczące w rozwijaniu systemu, pracownicy Działu Aplikacji Komputerowych (DAK) oraz Biura Spraw Studenckich (BSS) UW.

Ponieważ system jest cały czas rozwijany (w szczególności jeden z projektantów administruje produkcyjną bazą danych), jedynym akceptowalnym rozwiązaniem było włączenie się niektórych członków zespołu w prace wdrożeniowe. Jednak główny ciężar wdrożenia musiał przejąć kto inny. Zadanie to przypadło pracownikom DAK, którzy są odpowiedzialni za:

- koordynację przebiegu wdrożenia;
- instalowanie oprogramowania na stacjach klienckich i kontakty z wydziałami w sprawie konfiguracji i aktualizacji oprogramowania;
- opracowywanie procedur konwersji danych ze starych baz do bazy USOS oraz przeprowadzenie konwersji;
- ustalanie sposobu kodowania obiektów przechowywanych w bazie, który zapewni ich unikatowość, a zarazem ułatwi posługiwanie się nimi. Niektóre słowniki powinny zostać przeniesione z istniejącej bazy finansowo-księgowej (np. kody funkcji i stanowisk), inne wymagają zdefiniowania na potrzeby całej uczelni,

zgodnie z obowiązującymi przepisami MEN (np. słownik kierunków), jeszcze inne mają znaczenie ogólnopolskie (np. słownik powiatów i województw);

Zadania wymagające znajomości spraw studiów są realizowane przez pracownika BSS:

- szkolenie użytkowników w zakresie używania i konfigurowania systemu,
- pisanie dokumentacji użytkowej.

Ponieważ zarówno członkowie zespołu wdrożeniowego, jak i użytkownicy systemu pracują w tym samym przedsiębiorstwie, więc zasadniczo realizują wspólne cele. Świadomość tego powinna ułatwić osiągnięcie końcowego sukcesu.

### 3.3 Etapy wdrożenia

Ponieważ jednostki uczelni cieszą się dużą autonomią, to one muszą oficjalnie wyrazić chęć przystąpienia do wdrożenia. Trzeba przyznać, że zainteresowanie wdrożeniem jest bardzo dużo i na chwilę obecną przekracza możliwości zespołu wdrożeniowego (co oznacza, że wydziały muszą czekać w kolejce).

Pierwszy etap wdrożenia dotyczył wszystkich wydziałów i objął opracowanie wspólnych słowników dla istniejącej bazy finansowo-księgowej i dla USOS. Zasadniczo to USOS musiał się dostosować do większości już obowiązujących kodów, choć przy okazji udało się poprawić, ujednoczyć i uporządkować słowniki w starej bazie (wiele w nich było błędów literowych, czy niekonsekwentnych zasad kodowania). Niektóre decyzje wymagały akceptacji najwyższych władz (zatwierdzenia przez rektora), jak np. słownik struktury organizacyjnej jednostek UW. Te sprawy już od jakiegoś czasu czekały na uporządkowanie, ale potrzeby oprogramowania przyspieszyły działania.

Kolejny etap dotyczy już poszczególnych wydziałów. Wdrożenie rozpoczyna się od pobrania danych o pracownikach wydziału z bazy finansowo-księgowej. Dany ten są wyprowadzane na plik tekstowy, przeglądane, uzupełnianie i poprawiane. Następnie przy użyciu odpowiedniej procedury w SQL Plus wciąga się je do bazy produkcyjnej USOS.

W międzyczasie pracownik DAK zakłada na wydziale pierwsze stanowiska klienckie (z dostępem do bazy testowej), a pracownik BSS przychodzi na Wydział w celu odbycia pierwszych szkoleń z przyszłymi użytkownikami systemu. Z chwilą wciągnięcia danych pracowniczych wydziału do bazy produkcyjnej, zakłada się w bazie USOS pierwsze konta dla pracowników administracji Wydziału, definiuje dla nich również role i filtry. Przyszli użytkownicy przechodzą ostatnie szkolenie, już na bazie produkcyjnej, i podpisują zobowiązanie w sprawie ochrony danych osobowych. Od tej chwili stają się pełnoprawnymi (w ramach zdefiniowanej dla nich roli) użytkownikami USOS.

Role i filtry wymagają pewnego wyjaśnienia. Sprawą niezmiernie ważną jest opracowanie i wdrożenie właściwej polityki bezpieczeństwa, a co za tym idzie zdefiniowanie uprawnień (ról) użytkowników systemu przy pomocy dostępnych w USOS mechanizmów. Jest to problem zarówno organizacyjny, jak i informatyczny (USOS udostępnia mechanizmy pozwalające na bardzo precyzyjne definiowanie uprawnień, na poziomie wierszy i kolumn tabel bazy danych, ale jest to złożone zadanie informatyczne). Role te są więc definiowane przyrostowo, wraz z pojawianiem się w bazie kolejnych

porcji danych i kolejnych potrzeb (można np. nie nadawać uprawnień do obsługi spraw stypendialnych dopóty, dopóki wydziały nie zgłoszą potrzeby dostępu do takich danych). Na szczęście podsystem ról dostarcza wygodne mechanizmy kopiowania uprawnień, co pozwala rolę zdefiniowaną dla jednego wydziału skopiować dla innych wydziałów.

Sprawą istotną jest również przygotowanie spersonalizowanych zestawów filtrów pozwalających na wygodne poruszanie się użytkowników po olbrzymim wspólnym repozytorium danych. Filtry są także definiowane przyrostowo, najpierw dla jednego wydziału, a następnie kopiowane dla pozostałych.

Kolejny etap wdrożenia to wprowadzanie do bazy produkcyjnej danych o studentach. Dane te są bądź przegrywane z lokalnych baz wydziałów (w sposób podobny do przegrywania danych pracowniczych — zadanie jest jednak w tym wypadku trudniejsze, gdyż bazy wydziałów różnią się zawartością i sposobem przechowywania danych). Osobną procedurę przygotowano na potrzeby zgrywania danych nowo przyjętych studentów z bazy aplikacji wspomagającej Centralnej Rekrutacji Kandydatów na studia.

Równolegle odbywa się proces definiowania dydaktycznej oferty wydziałów. Obejmuje ona informacje o prowadzonych przez wydział kierunkach i programach studiów oraz o możliwych do zdobycia certyfikatach. Z każdym etapem programu studiów wiążą się wymagania, które trzeba spełnić żeby zaliczyć ten etap. Zazwyczaj pełna informacja na ten temat znajduje się już w informatorze o studiach, musi jednak zostać przetłumaczona na logikę systemu USOS. Jest to trudne i żmudne zadanie, wymagające intensywnego zaangażowania wydziałowych ekspertów do spraw studiów, na szczęście na pełną jego realizację jest sporo czasu. Żeby wprowadzić dane o studentach, wystarczy zdefiniować kierunki i programy. Pełny opis wymagań będzie potrzebny z chwilą, gdy wydział przystąpi do rozliczania studentów. Nieco prostszym zadaniem jest wprowadzenie informacji o oferowanych przedmiotach i ich atrybutach. Dwa lata temu powstał prototypowy ogólnouniwersytecki katalog przedmiotów. Na jego potrzeby wydziały opracowały system kodów oraz przygotowały opis programów w postaci plików Excela. Większość z tych danych da się bez trudu przegrać do bazy USOS.

Opisane czynności wymagać będą pewnych kompromisów, dostosowania się do obowiązującej struktury bazy. Będzie się to wiązało z ujednocinaniem i upraszczaniem zasad i procedur. Będzie to długi proces, ale równocześnie niewątpliwie korzystny dla społeczności uniwersyteckiej. Z jednej strony przeszkodą jest autonomia jednostek, z drugiej ujednocinaniu powinien sprzyjać rachunek ekonomiczny. Przykładem może być rezygnacja na Wydziale MIM z kart egzaminacyjnych czy zmiana procedury skreśleń studenta — nowe raporty przygotowane na potrzeby jednego wydziału będzie można zastosować i na pozostałych, o ile zdecydują się one na wdrożenie analogicznych procedur organizacyjnych.

Sprawą ważną są częste spotkania z użytkownikami i informowanie ich na bieżąco o wszystkich planach. W tym celu stworzono specjalną listę dyskusyjną i serwis WWW.

Szczęśliwie przebieg wdrożenia podlega umiarkowanej presji czasu. Wiadomo co prawda, że niektóre etapy wdrożenia muszą być zgodne z przebiegiem roku akademickiego, jednak pewne przesunięcia są akceptowalne i nie narażają powodzenia całego procesu. Również opóźnienia w dostarczaniu nowych modułów nie powodują

nieodwracalnych skutków finansowych czy organizacyjnych. To pozwala na poświęcenie uwagi na unifikację słowników, wypracowanie dobrych sposobów kodowania oraz lepszą organizację procedur. Celem jest pełne zintegrowanie procesu obsługi spraw studiów wokół wspólnej ogólnouniwersyteckiej aplikacji i wspólnego repozytorium danych.

## 4. Podsumowanie

Miejsce powstawania Uniwersyteckiego Systemu Obsługi Studiów ma istotne znaczenie dla trybu prowadzenia projektu i jego wdrożenia. Mając zobowiązania dydaktyczne wobec studentów uczestniczących w projekcie, trzeba przykładać większą wagę do stosowanych metod i narzędzi inżynierii oprogramowania. Oczywiście, jak to w każdym projekcie programistycznym, użytkownik chce dostawać nowe moduły jak najszybciej. Ale w przypadku tego projektu cel dydaktyczny jest nadrzędną wartością i nie może być przedmiotem żadnych kompromisów. Musi się znaleźć czas na eksperymentowanie z nowymi metodami, narzędziami i technikami inżynierii oprogramowania, trzeba umieć podejmować decyzje o wycofaniu z użycia modułu, który nie spełnia kryteriów jakościowych i napisaniu jego nowej wersji. Jako dydaktyk inżynierii oprogramowania jestem jednak przekonana, że to takie podejście może mieć jedynie pozytywny wpływ na powstające oprogramowanie. Nagrodą są z jednej strony zadowoleni użytkownicy, z drugiej zaś studenci, którzy, mając także pewne doświadczenia z zewnętrznymi firmami programistycznymi, doceniają dydaktyczny walor uczestniczenia w projekcie, które daje im możliwość przyswajania dobrych praktyk.

Wdrożenie przeprowadzane własnymi siłami pozwala na stopniowe wprowadzanie systemu do użytku, lepsze dostosowanie go do potrzeb użytkowników, a tempa wdrożenia do możliwości wydziałów,

## Bibliografia

- [1] Astels D., Miller G., Novak M., *A Practical Guide to Extreme Programming*, Prentice Hall, 2002.
- [2] Goldman, B., *Uniwersytecki System Obsługi Studiów. Standardy*. Instytut Informatyki Uniwersytetu Warszawskiego. Praca magisterska (w przygotowaniu).
- [3] Gylseth S., *Using CDM Fast Track, Oracle's DSDM Compliant RAD Approach*, Oracle Corporation, 2000.
- [4] Karwański, M., *Uniwersytecki System Obsługi Studiów. Doktoranci*. Instytut Informatyki Uniwersytetu Warszawskiego. Praca magisterska, 2001.
- [5] Mincer-Daszkiewicz, J., *Tworzenie produkcyjnego oprogramowania w środowisku akademickim*, III Krajowa Konferencja Inż. Oprogramowania – KKIO'2001, 2001, 225–236.
- [6] Sommerville, I., *Software Engineering (wydanie 6)*. Addison-Wesley, 2000.
- [7] Sroka, J., *Uniwersytecki System Obsługi Studiów. Akademiki*. Instytut Informatyki Uniwersytetu Warszawskiego. Praca magisterska (w przygotowaniu).
- [8] Strona domowa projektu USOS, Wydział MIM UW, <http://usos.mimuw.edu.pl>.
- [9] TOAD, Tool for Oracle Application Developers, Quest Software.