

**Bartosz JABŁOŃSKI**

Politechnika Wrocławska, Polska

## **Uwzględnianie oczekiwań pracodawców w definiowaniu zakresu przedmiotowego kursów na przykładzie nauczania platform programistycznych**

### **Wstęp**

W wielu dziedzinach nauczania obserwuje się coraz silniejszą potrzebę aktualizacji programów nauczania do realiów rynkowych. Praktyczna odpowiedź na tego rodzaju zapotrzebowanie realizowana jest na różnych poziomach – zaczynając od Krajowych Ram Kwalifikacji wynikających z przyjętych założeń deklaracji bolońskiej, poprzez ich wdrażanie w kolejnych szkołach wyższych, a kończąc na tworzeniu indywidualnych programów nauczania.

Ze względu na wzrastające zapotrzebowanie wykwalifikowanych kadr obserwuje się coraz większy nacisk na podnoszenie jakości kształcenia w szczególności na kierunkach technicznych, takich jak informatyka lub pokrewne. Jednocześnie obszar IT jest jedną z najbardziej dynamicznie rozwijających się i zmieniających dziedzin. Przykładowo według [Bureau 2012] rynek programistów rozwijać się będzie w ciągu najbliższych lat w tempie 22% przy średnim rozwoju wszystkich branż na poziomie 8%.

To powoduje, że stosowane do tej pory programy oraz metody nauczania nie będą dłużej wystarczające, aby móc skutecznie odpowiedzieć na zapotrzebowanie pracodawców. Programy nauczania, które nie będą adaptacyjnie uwzględniać nowych trendów i pojawiających się technologii, nie będą mogły odpowiedzieć na rynkowe zapotrzebowanie oraz nie zapewnią odpowiedniego przygotowania dla studentów.

Istnieją różne sposoby tworzenia środowiska współpracy pomiędzy uczelniami oraz przedstawicielami przemysłu. Wśród nich można zaobserwować ewaluację wyników badań przemysłowych w środowisku akademickim [Ligus 2013] lub organizowanie przestrzeni do spotkań pomiędzy tymi dwoma środowiskami ze szczególnym śledzeniem losów absolwentów uczelni [Poulova 2013]. Problem dopasowania oferty edukacyjnej do rynku pracy jest omawiany w szerszym zakresie w pracy [Kozieł 2012].

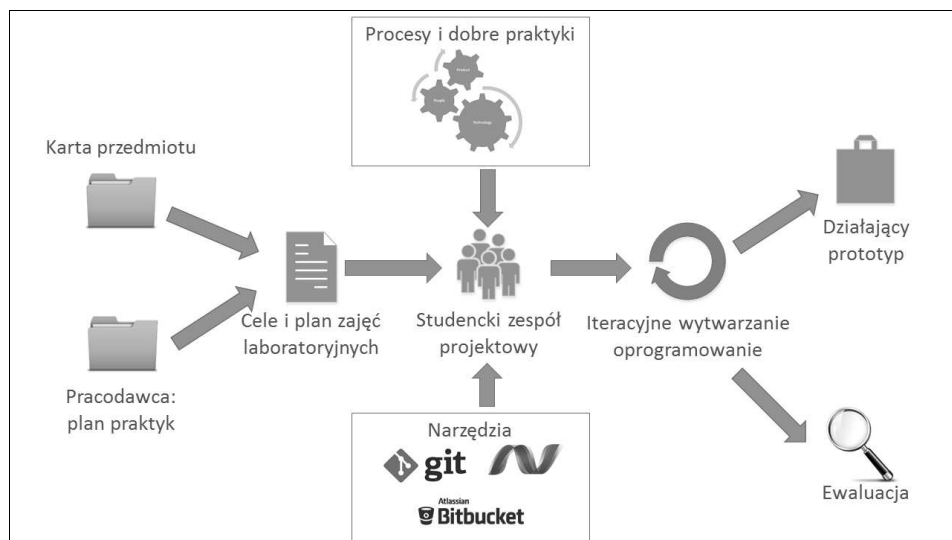
Często spotykanym do tej pory podejściem w nauczaniu podstaw programowania jest definiowanie ćwiczeń praktycznych dla wyodrębnionych problemów i pojęć, takich jak: poznawanie podstawowych konstrukcji języka (iteracyjność, rekurencja), proste konstrukcje logiczne, podstawy struktur danych.

Podejście to jest oczywiście potrzebne i w wielu przypadkach niezbędne, jednak jest ono również niewystarczające w obliczu bardzo dynamicznych zmian technologicznych.

Jednym z przedmiotów prowadzonych na Wydziale Elektroniki Politechniki Wrocławskiej jest kurs „Platformy programistyczne .NET i Java”. W ramach tego kursu w 2014 r. wykorzystano nowe podejście do opracowania i realizacji szczegółowego programu zajęć. W prezentowanym artykule zostaną przedstawione przyjęte założenia oraz zaproponowany, a następnie zaimplementowany proces nauczania wraz z wykorzystanymi narzędziami dydaktycznymi. Omówione zostaną rezultaty ewaluacji kursu.

### Proces przygotowania zakresu materiału i realizacji kursu

Aby możliwe było odpowiedzenie na zapotrzebowania rynkowe z punktu widzenia umiejętności przyszłych absolwentów, niezbędne jest wykorzystanie zróżnicowanych źródeł danych. Podobna potrzeba, a także podobne rozwiązanie zostało zaprezentowane w pracy [Matusa 2013]. W omawianym przypadku proces przygotowania zakresu materiału kursu, a następnie jego realizacji został zobrazowany na rys. 1.



Rys. 1. Proces przygotowania i realizacji kursu platformy programistyczne

Podstawą programową realizowanego kursu jest zatwierdzona karta przedmiotu definiująca cele oraz przedmiotowe efekty kształcenia. Aby jednak możliwe było stworzenie szczegółowego planu zajęć laboratoryjnych, zakres tematyczny został skonfrontowany z programem praktyk jednej z firm programistycznych we Wrocławiu. Przykładowo:

- Karta przedmiotu definiuje, że student potrafi napisać prostą aplikację okienkową na platformie .NET.
- Plan praktyk firmy programistycznej zakłada udział w projektowaniu bazy danych dla aplikacji.
- Na tej podstawie w planie zajęć laboratoryjnych przyjęto, że zostanie opracowany projekt i implementacja konkretnej aplikacji, która m.in. wykorzystuje bibliotekę EntityFramework wraz z podejściem Code First do realizacji tego zadania.

Na podstawie analizy zakresu przedmiotu oraz oczekiwań przyszłych pracodawców zdefiniowano kilka zestawów zadań w ramach laboratoriów:

- ćwiczenia wstępne – praktyczne wykorzystanie podejścia typu Test Driven Development,
- ćwiczenia warsztatowe – tworzenie prostego systemu symulacyjnego,
- ćwiczenia projektowe – iteracyjne tworzenie systemu decyzyjnego składającego się z kilku modułów realizowanych w trakcie kolejnych zajęć. System taki mógł być rozpatrywany jako prototyp prawdziwej aplikacji komercyjnej i wymagał wykorzystania zróżnicowanych bibliotek dodatkowych (np. połączenia sieciowe HTTP, połączenie do API, obsługa formatu JSON, komunikacja z bazą danych).

Aby zaznajomić studentów z komercyjnie stosowanym procesem wytwórczym oprogramowania, wprowadzono zestaw praktyk dodatkowych:

- Obowiązkowe wykorzystanie repozytorium kodu (np. Git) jako podstawowej platformy do przechowywania i współdzielenia rezultatu kolejnych ćwiczeń.
- Współpraca grupowa na zasadzie programowania w parach.
- Przeglądy kodu (*code review*) z wykorzystaniem platformy Bitbucket. W ramach przeglądów kodu kładziony był nacisk na: poprawność zrealizowanej funkcjonalności (rozumienie specyfikacji wymagań wirtualnego „klienta”), przejrzystość kodu, dobre rozumienie i stosowanie obiektowości itd.
- Retrospekcje – regularne, grupowe podsumowanie osiągniętych celów oraz określanie obszarów, w których skuteczność realizacji aplikacji może zostać poprawiona. Jest to podejście stosowane typowo w zwinnych środowiskach wytwórczych, np. Scrum [Kniberg 2007]. Pozwalało to również na konstruktywną wymianę doświadczeń pomiędzy grupami projektowymi i naukę szukania nowych rozwiązań.

Iteracyjny rozwój aplikacji przy jednoczesnym określeniu pożądanego produktu docelowego pozwolił osiągnąć następujące efekty:

- zapoznanie się z wybraną platformą całościowo w ścisłym powiązaniu z realizowanym celem projektowym,
- poznanie dobrych praktyk i narzędzi stosowanych w komercyjnych procesach wytwórczych (np. Test Driven Development),
- pobudzenie rzeczywistej współpracy pomiędzy osobami w grupie i rozwój umiejętności komunikacyjnych,

- praktyczne wykorzystanie podstawowych pojęć programistycznych (np. projektowanie obiektowe) i rozwinięcie umiejętności ich wykorzystania bezpośrednio w rzeczywistym, potencjalnie komercyjnym zastosowaniu,
- rozwinięcie umiejętności rozumienia specyfikacji oraz praktycznego poszukiwania nowych, nieznanych wcześniej informacji (np. bibliotek dodatkowych).

## Ewaluacja

Po zakończonym kursie przeprowadzono anonimową ankietę ewaluacyjną mającą na celu sprawdzenie poziomu satysfakcji uczestników. Otrzymane rezultaty zostały przedstawione w tabeli 1. W ankiecie wzięło udział 20 uczestników.

**Tabela 1**

### Rezultaty ankiety ewaluacyjnej

	Średnia	Odch. Stand.	Wsp. Zmienności
Jak ogólnie oceniasz kurs Platformy programistyczne .NET i Java?	4,84	0,36	7,5%
Jak oceniasz sposób prezentacji materiału na wykładzie?	4,63	0,58	12,6%
Jak oceniasz sposób przerabiania materiału na laboratorium?	4,58	0,59	12,9%

Przedstawione wyniki pokazują niskie zróżnicowanie otrzymanych ocen przy bardzo wysokiej wartości ocen (przyjęta została skala: 1 – bardzo słabo, 3 – średnio, 5 – bardzo dobrze).

Wśród otrzymanych komentarzy dodatkowych warto wymienić powtarzające się głosy o zaletach tego rodzaju podejścia:

- fakt tworzenia aplikacji o potencjalnym zastosowaniu praktycznym,
- wykorzystanie wielu bibliotek,
- wykorzystanie realnych narzędzi (np. Git),
- możliwa samodzielność w znajdowaniu praktycznego rozwiązania,
- określenie celu finalnego, do którego należy zmierzać przez realizację kolejnych iteracji.

## Podsumowanie

Bardzo szybki rozwój technologiczny oraz duże zapotrzebowanie na dobrze wyszkolonych absolwentów wymaga coraz bliższej współpracy pomiędzy uczelniami i przedstawicielami przemysłu. W omawianym w artykule przypadku zostało zaprezentowane podejście zastosowane w trakcie jednych z zajęć prowadzonych na Politechnice Wrocławskiej.

W ramach tworzenia szczegółowego programu zajęć wykorzystano zarówno z programu praktyk przedsiębiorstwa programistycznego, jak i dobrych praktyk, narzędzi i wzorców wytwarzania oprogramowania komercyjnego. Pozwoliło to na całościowe zapoznanie się z platformami programistycznymi będącymi przedmiotem kursu. Jednocześnie studenci mogli w praktyce przećwiczyć techniczne umiejętności programistyczne oraz niezbędne umiejętności dodatkowe (rozumienie specyfikacji, tworzenie wielokrotnie używalnego kodu, komunikacja zespołowa, dbanie o techniczną jakość rozwiązania).

Zastosowane podejście spotkało się z bardzo dobrym przyjęciem ze strony studentów – przeprowadzona ewaluacja pokazuje, że satysfakcja z zajęć była na bardzo wysokim poziomie. W dalszych krokach planowane są badania nad skutecznością przygotowania studentów do praktyk lub pracy komercyjnej w przedsiębiorstwie.

## Literatura

Bureau of Labor Statistics (2012): *Occupational Outlook Handbook*, United States Department of Labor.

Kniberg H. (2007): *Scrum and XP from the Trenches*, InfoQ.

Matusa R., Butkus L., Krilavicius T., Man K.L., Hai-Ning Liang (2013): *Improving the Teaching of Computer Networks through the Incorporation of Industry Based Training Courses*, [w:] Proc. Of the 2013 IEEE International Conference *Teaching, Assessment and Learning for Engineering (TALE)*.

Ligus J., Ligusova J., Gazda T. (2013): *Cooperation between Industry and University Based on the Evaluation of the Industrial Research Results in the Academic Environment*, [w:] 2013 Proceedings of the 24th EAEEIE Annual Conference (EAEEIE).

Poulova P., Klimova B.F. (2013): *Linking Theory with Practice*, [w:] *Proc. Of the 2013 International Conference on Information Technology Based Higher Education and Training (ITHET)*.

Koziel G. (2012): *University Education Tailored to Labour Market Expectations*, [w:] *Global Engineering Education Conference (EDUCON), 2012 IEEE*.

## Streszczenie

W opracowaniu został zaprezentowany jeden ze sposobów uwzględniania aktualnej wiedzy w procesie kształcenia oferowanego w szczególności dla kierunków technicznych. Zaprezentowana została analiza przypadku dla prowadzonego na Politechnice Wrocławskiej kursu „Platformy programistyczne .Net i Java”. Uwzględnienie potrzeb przyszłych pracodawców w programie zajęć kursu pozwoliło osiągnąć wiele pozytywnych rezultatów: skupienie się na umiejętnościach niezbędnych dla absolwenta, poznanie aktualnych całościowych rozwiązań technologicznych oraz mierzalny wysoki poziom satysfakcji uczest-

ników kursu. W artykule przedstawiono również przykładowe wykorzystane praktyki, które mogą być z powodzeniem stosowane także w przypadku innych przedmiotów i kierunków.

**Słowa kluczowe:** dopasowanie oferty edukacyjnej, współpraca z przemysłem, iteracyjny rozwój oprogramowania, platformy programistyczne.

## **Taking Into Account the Expectations of Employers in Defining the Scope of the Courses on the Example of Teaching Development Platforms**

### **Abstract**

This paper presents one of the methods of taking into consideration current knowledge in the educational process, specifically for technical departments. Case study results are presented for a course provided within Wrocław University of Technology „Programming platforms .Net and Java”. Taking into consideration needs of future employers helped dramatically to reach several goals: focusing on skills and knowledge, which are crucial for graduates, having contact with up to date technical solutions and high level of participants satisfaction. In this paper a list of good practices were presented – they can be certainly used in cases for other courses and field of studies.

**Keywords:** education scope tailoring, industrial cooperation, iterative software development, programming platforms.