

NAUKA PROGRAMOWANIA W SZKOŁACH

Czas na upgrade
– perspektywa 2014



CENTRUM
CYFROWE

projekt:polska®



SZEROKIE POROZUMIENIE
NA RZECZ UMIEJĘTNOŚCI
CYFROWYCH W POLSCE



SPIS TREŚCI

3_Wstęp

5_Wyzwania dla nauczania programowania w szkołach

9_Rekomendacje – jak wspierać *upgrade* nauki programowania w szkołach

13_Mistrzowie kodowania – programowanie jako krok ku nowoczesnej i otwartej szkole

Autorzy raportu:

Anna Mazgal, Alek Tarkowski

Autorzy ewaluacji:

Mateusz Halawa, Paulina Wróbel

Projekt graficzny:

Paulina Tyro-Niezgoda



Warszawa, styczeń 2014



Raport jest dostępny na licencji Creative Commons Uznanie autorstwa 3.0 Polska. Pewne prawa zastrzeżone na rzecz autorów i Centrum Cyfrowego Projekt: Polska. Treść licencji jest dostępna na stronie <http://creativecommons.org/licenses/by/3.0/pl/deed.pl>

Zdjęcie na okładce: lekcja dla Mistrzów Kodowania w Szkole Podstawowej nr 5 w Skarżysku Kamiennej, fot.: materiały organizatora

WSTĘP

W niniejszym dokumencie pragniemy rozwinąć kwestie przedstawione w opublikowanym w 2012 roku raporcie *Nauka programowania w szkołach – czas na upgrade?*, będącym raportem otwarcia programu „Mistrzowie kodowania”. Przedstawimy szerszy kontekst nauki programowania w szkołach, związany w szczególności z wymiarem strategicznym tych działań oraz odpowiednim kształtowaniem polityki publicznej związanej z edukacją cyfrową.

Dokument ten składa się z trzech części. W części pierwszej prezentujemy wyzwania strategiczne związane z wdrażaniem nauczania programowania w szkołach. Część druga zawiera rekomendacje, które jak mamy nadzieję posłużą za inspirację w odpowiadaniu na zidentyfikowane wyzwania. Część trzecia to swoisty materiał dowodowy – na przykładzie ewaluacji pilotażu programu „Mistrzowie kodowania” pokazujemy, jak uczenie programowania w oparciu o idee otwartości, współpracy oraz zniesienie paradygmatu „mistrz-uczeń” przybliży nas do nowoczesnej szkoły. A więc takiej, która uczy rozwiązywania problemów, buduje kompetencje społeczne oraz przygotowuje młodzież do mierzenia się z wyzwaniami rynku pracy.



WYZWANIA DLA NAUCZANIA PROGRAMOWANIA W SZKOŁACH

PROGRAMOWANIE JAKO WYZWANIE RYNKU PRACY

Raport *Nauka programowania w szkołach – czas na upgrade?* podkreśla, że zwiększone zapotrzebowanie na pracowników posiadających umiejętności programowania jest jednym z najważniejszych wyzwań rynku pracy. Jest ono dostrzegane na poziomie wspólnotowym i stało się ważnym elementem polityk UE.

Grand Coalition for Digital Jobs to wielosektorowe partnerstwo na rzecz zwiększenia zatrudnienia w zawodach związanych z technologią, zainicjowane przez Prezydenta UE José Manuela Barroso w marcu 2013 r. Według Koalicji, siła robocza w sektorze związanym z technologiami informacyjno-komunikacyjnymi (ICT) stanowiła w 2011 r. w Europie 6,67 miliona osób, co stanowi 3,1 procent europejskiej siły roboczej. W niektó-

rych krajach ludzie ci mogliby stanowić 6 procent siły roboczej¹. Komisja Europejska podaje, że do 2015 r. brakować będzie 900 tys. profesjonalistów w tej dziedzinie.

Jednocześnie programowanie to nie tylko kwestia znalezienia dobrego zatrudnienia w zawodzie związanym z ICT. Jak podaje nasz wcześniejszy raport, programowanie to dzisiaj umiejętność o znaczeniu horyzontalnym, potrzebna w wielu innych zawodach. Choćby wiele zawodów kreatywnych – od artystów przez projektantów graficznych po projektantów przemysłowych – wymaga przynajmniej podstawowej znajomości programowania. Znaczenie tych umiejętności będzie zatem rosnąć wraz ze wzrostem wykorzystania technologii cyfrowych we wszystkich sektorach gospodarki.

PROGRAMOWANIE JAKO WYZWANIE KOMPETENCYJNE

Zapewnienie odpowiednio wykształconych kadr gospodarce wymaga dostarczenia uczącym się odpowiednich kompetencji. Wyzwaniem jest nie tylko umiejscowienie nauki programowania w systemie edukacji formalnej i nieformalnej, ale także wyważenie na ile stawiamy na teoretyczną wiedzę programistyczną, a jak ważne są praktyczne umiejętności. Pierwsza będzie ważniejsza dla ekspertów - informatyków; drugie - istotne dla nabywania powszechnych kompetencji.

Komisja Europejska przyjęła następujące rozróżnienie umiejętności uwzględniające potrzeby nabywających je osób:

Umiejętności praktyka (ICT practitioner skills) – są to umiejętności potrzebne do bada-

Wyzwania dla nauczania

nia, rozbudowywania, projektowania, planowania strategicznego, zarządzania, wytwarzania, konsultingu, marketingu, sprzedaży, integrowania, instalacji, administrowania, utrzymywania, wsparcia i serwisowania systemów ICT.

Umiejętności użytkownika (ICT user skills)

– te są wymagane do efektywnego korzystania z systemów i urządzeń ICT przez jednostkę. Użytkownik ICT korzysta z technologii i narzędzi ICT w swojej pracy. Umiejętności użytkownika obejmują użycie popularnego oprogramowania, ale także tego specjalistycznego, wykorzystywanego w konkretnym zawodzie.

Umiejętności e-biznesowe (e-Business skills lub e-leadership skills)

– konieczne do wykorzystywania możliwości, jakie stwarza ICT, a zwłaszcza internet, oraz do zapewnienia skuteczności i wydajności różnych typów organizacji z tych technologii korzystających. Także zdolność testowania nowych metod prowadzenia działalności biznesowej oraz procesów administracyjnych i organizacyjnych oraz zakładania nowych przedsiębiorstw wykorzystujących ICT².

Programowanie z pewnością wpisuje się najszerszej w model umiejętności praktyka, jednak rosnąca rola technologii sprawia, że i użytkownicy w coraz większej mierze będą musieli zdobywać tego rodzaju kompetencje, zwłaszcza w zawodach kreatywnych oraz w innowacyjnych przedsiębiorstwach, gdzie dostępne oprogramowanie

może nie być dopasowane do nowych lub specyficznych potrzeb.

Samo pisanie kodu nie wystarczy, jeśli nie towarzyszą mu kompetencje związane ze zrozumieniem informacji i logiką rozwiązywania problemów. Należy łączyć, na wyższych poziomach edukacji, umiejętności programistyczne na przykład z kompetencjami biznesowymi. Mapowanie tych kompetencji w projektowaniu systemu nauczania jest wyzwaniem strategicznym. W szczególności należy wpisać kompetencje związane z programowaniem w strukturę kompetencji medialnych bądź cyfrowych.

Warto wskazać na przykład Wielkiej Brytanii, w której dodając naukę programowania do podstawy programowej postanowiono przeorientować nauczanie kwestii związanych z technologiami cyfrowymi. Odchodząc od terminu „ICT” uznano umiejętność programowania komputerów (*computing*) za kluczowe pojęcie organizujące edukację w tym obszarze. Taka zmiana byłaby w Polsce dziś przedwczesna, przykład brytyjski wskazuje jednak możliwy kierunek rozwoju uczenia kompetencji cyfrowych w szkołach, wraz z popularyzacją nauki programowania.

PROGRAMOWANIE JAKO WYZWANIE W SYSTEMIE KSZTAŁCENIA – NIERÓWNOŚCI SPOŁECZNE

W systemie kształcenia nauka programowania napotyka na przeszkody charakterystyczne i dla

innych przedmiotów, w tym te związane z niewystarczającą infrastrukturą doksztalcenia nauczycieli i obudowania zajęć nowoczesnymi pomocami dydaktycznymi - jednak spotęgowane tym, że jest to zagadnienie stosunkowo nowe. Nie można go rozpatrywać w oderwaniu od dostępnej infrastruktury – czy to komputerów w szkołach, czy to zajęć dostępnych w ramach programów kształcenia nieformalnego, czy wreszcie w powiązaniu z dostępnością prywatnych komputerów czy urządzeń mobilnych.

Nauka programowania jest wyzwaniem w kontekście nierówności społecznych, które są barierą w dostępie do edukacji. W raporcie *Nauka programowania w szkołach – czas na upgrade?* podkreślamy, że rolą szkoły jest zmniejszanie tego rodzaju nierówności. Jednak samo to stwierdzenie nie będzie w żadnej mierze konstruktywne, jeśli stosowne działania nie będą uwzględnione na poziomie systemowym. Można mieć nadzieję, że kontynuacja, po etapie pilotażu, programu „Cyfrowa szkoła” doprowadzi do tego rodzaju systemowych skutków.

Innym rodzajem nierówności, w którego zniesieniu istotną rolę ma szkoła, jest niewystarczający udział dziewcząt w kształceniu ścisłym, a co za tym idzie w nauce programowania. Przedmioty te nadal są tradycyjnie uznawane za domenę chłopców, co ostatecznie wpływa na wybór dalszej ścieżki kształcenia i kariery – obecnie kobiety stanowią niecałe 30 procent zatrudnionych w ICT⁴. Zmiana postawy jest procesem dłu-

gotrwałym, jednak w projektowaniu kształcenia uwzględniającego programowanie trzeba pamiętać o tym zjawisku i konstruować odpowiedni przekaz.

Ma to bezpośrednie znaczenie dla gospodarki, gdyż jak podaje raport *Women active in the ICT sector*, przygotowany w 2013 r. dla Komisji Europejskiej, więcej kobiet w zawodach związanych z technologiami cyfrowymi przyniosłoby rocznie dodatkowe 9 miliardów PKB w skali Unii Europejskiej⁵. Ma to też znaczenie ekonomiczne dla potencjalnych pracowniczek, gdyż są to dobrze wynagradzane stanowiska.

PROGRAMOWANIE JAKO WYZWANIE DLA NAUCZYCIELI

Jako że nauka programowania jest tematem nowym w edukacji powszechnej, nauczycielom często brak kompetencji koniecznych dla zapewnienia dostępu do tego rodzaju edukacji. Problemy związane z nauczaniem programowania będą jeszcze większe niż w wypadku innych form wykorzystania ICT w procesie uczenia – które także natrafiają na barierę w postaci niewystarczających kompetencji nauczycieli.

Jest to problem ogólnoeuropejski, wychwycony również przez Komisję Europejską w badaniu *ICT in Education* w którym zebrano i opracowano dane z 31 państw Europy (27 członkowskich oraz Chorwacji, Islandii, Norwegii i Turcji) na temat dostępu, użytkowania, kompetencji oraz postaw

uczniów i nauczycieli dotyczących nowych technologii w szkole⁶.

Rekomendacje zawarte w dokumencie to konkretne, „twarde” metody wsparcia nauczycieli na poziomie systemowym. Są to dwa zasadnicze działania:

1. Konieczne jest zintegrowane podejście do nauczania ICT w szkołach, oznaczające nie tylko inwestycje w infrastrukturę, ale także większe nakłady na kształcenie nauczycieli, nagrody za korzystanie z ICT na lekcjach, a także tworzenie stanowisk koordynatora ds. ICT. Tego rodzaju wsparcie jest dziś kluczowym elementem polskiego programu „Cyfrowa szkoła”.
2. Na poziomie wspólnotowym zaleca się redukcję rozbieżności między edukacją w zakresie ICT dla nauczycieli w poszczególnych krajach UE, wsparcie dla projektów wprowadzających nauczanie poprzez zastosowanie nowych technologii, wsparcie dla tworzenia wysokiej jakości materiałów do nauczania i regularne monitorowanie wykorzystywania zarówno technologii jak i kompetencji cyfrowych.

Oba te postulaty w praktyce sprowadzają się do zwiększenia nakładów finansowych – czy to na poziomie wspólnotowym, czy to krajowym – na infrastrukturę wspierającą nauczycieli w rozwoju zawodowym i dostarczającą im właściwych narzędzi do przekazywania swojej wiedzy nowym

pokoleniom. Jakkolwiek jest to podejście słuszne, nie jest ono łatwe do wypracowania ani w krótkim okresie, ani w sytuacji ograniczonych zasobów na oświatę w ogóle.

Podstawą kształcenia nauczycieli jest odpowiednia edukacja pedagogiczna na uczelniach. Programy podyplomowe, wdrażane na uczelniach w pierwszej dekadzie XXI wieku, nie są już obecnie prowadzone. W programie „Cyfrowa szkoła” ciężar doszkalania nauczycieli spoczywa niemal całkowicie na organizacji pozarządowej - Centrum Edukacji Obywatelskiej. Ta cenna forma współpracy międzysektorowej z kompetentnym partnerem dowodzi istnienia deficytów w instytucjach publicznych odpowiedzialnych za kształcenie nauczycieli.

SZKOŁA JAKO WYZWANIE DLA PROGRAMISTÓW

Raport *Nauka programowania w szkołach – czas na upgrade?* potwierdza powszechną zapewne intuicję, że szkoła nie jest również atrakcyjnym miejscem dla programistów. Wobec zapotrzebowania na te umiejętności w dobrze płatnych sektorach gospodarki trudno się dziwić, że programiści nie wybierają szkoły jako miejsca realizacji swoich zawodowych aspiracji.

Z pewnością sama umiejętność kodowania nie wystarczy, by dobrze uczyć programowania innych. Należy zatem zastanowić się nad sposobami angażowania programistów w taki sposób, by mo-

Wyzwania dla nauczania

gli oni dzielić się wiedzą, nie rezygnując ze swojej ścieżki zawodowej. Pod tym względem szkoły powinny czerpać wzór z inicjatyw oddolnych, takich jak Coder Dojo czy Geek Girl Carrots, zdolnych wytworzyć kulturę organizacyjną atrakcyjną dla programistów.

PROGRAMOWANIE JAKO WYZWANIE OTWARTEJ EDUKACJI

Jedną z rekomendacji raportu *Nauka programowania w szkołach – czas na upgrade?* dotyczyła oparcia nauki programowania o wartości i doświadczenia ruchu otwartej edukacji.

Otwarta edukacja to kształcenie z wykorzystaniem zasobów, które można swobodnie pobierać, wykorzystywać, adaptować i udostępniać innym w wersji oryginalnej lub przetworzonej w zasadzie bez ograniczeń. Efekt ten uzyskuje się poprzez połączenie otwartości technicznej – oznaczającej swobodną dostępność treści w Sieci – z otwartością prawną polegającą na odpowiednim licencjonowaniu zasobów.

Otwarcie zasobów edukacyjnych redukuje i racjonalizuje koszty edukacji po stronie instytucji oraz uczniów, wyrównując dostęp do wiedzy i zwiększając szanse edukacyjne. Nie bez znaczenia jest też nieograniczona możliwość aktualizacji, dostosowywania oraz łączenia treści edukacyjnych w indywidualne programy nauczania, czy ułatwiona personalizacja nauki.

Programowanie to korzystanie z otwartych, swobodnie dostępnych narzędzi i zasobów oraz swobodna wymiana wiedzy z innymi programistami. Języki programowania są z założenia dobrem wspólnym, tworząc system porozumiewania się między programistami oraz między ludźmi i maszynami. Tradycja otwartości w informatyce jest integralną częścią nauki programowania, a ta ma szansę stać się kluczową formą nowoczesnej, otwartej edukacji.

Jednym z większych wyzwań otwartej edukacji jest wypracowanie równie otwartych mechanizmów współpracy i wymiany wiedzy między nauczycielami oraz między uczniami. Pod tym względem programy nauki programowania, czerpiące z doświadczeń i kultury programistów, mogą być wiodącym przykładem całościowo rozumianej otwartej edukacji. Otwartość zasobu edukacyjnego jakim jest oprogramowanie szczególnie sprzyja współpracy i tworzeniu partnerstw międzysektorowych, ponieważ nie ma znaczenia kto je wytworzył, skoro mogą być potem wykorzystywane bez większych ograniczeń.

PROGRAMOWANIE JAKO WYZWANIE STRATEGICZNE DLA ROZWOJU KAPITAŁU SPOŁECZNEGO POLSKI

Jeśli uznamy naukę programowania za kluczowy wymiar działań związanych z ICT w szkołach, to należy uznać większość obecnych dokumentów strategicznych za anachroniczne.

Kwestie te nie są poruszane w żadnym z kluczowych i obowiązujących dzisiaj dokumentów strategicznych, począwszy od Długookresowej Strategii Rozwoju Kraju, przez Średniookresową Strategię Rozwoju Kraju, po zintegrowane strategie, takie jak Strategia Rozwoju Kapitału Ludzkiego czy Strategia Rozwoju Kapitału Społecznego. Nauka programowania nie jest wyróżniona jako kluczowa kompetencja, choć została uwzględniona w podstawie programowej. Należy zaznaczyć, że kompetencje związane z programowaniem nie są zmapowane w dostępnych dokumentach rekomendacyjnych przygotowanych przez organizacje, wliczając w to opracowany przez Fundację Nowoczesna Polska Katalog kompetencji medialnych i informacyjnych⁷.

Jedyne odniesienia do umiejętności programowania znajdują się w Programie Operacyjnym Polska Cyfrowa. Jeden z jego celów szczegółowych to „pobudzanie potencjału uzdolnionych programistów dla zwiększenia zastosowania rozwiązań cyfrowych w gospodarce i administracji” i związane z nim działanie „E–pionier – opracowywanie nowych rozwiązań na potrzeby społeczne i gospodarcze bazujących na TIK poprzez akcelerację pomysłów zdolnych programistów.” Jest to jednak działanie skierowane do osób o zaawansowanych kompetencjach programistycznych, na ostatnim szczeblu edukacji formalnej, jakim jest szkolnictwo wyższe. A to tylko jeden aspekt nauki programowania, zaplanowany niejako kosztem edukacji w zakresie kompetencji powszechnych.

REKOMENDACJE – JAK WSPIERAĆ UPGRADE NAUKI PROGRAMOWANIA W SZKOŁACH?

1. Szkoła jako miejsce rozwiązywania problemów

Programowanie to coś więcej niż tylko nauka poprawnego pisania kodu. Kodowanie powinno być poligonem doświadczalnym dla nowoczesnego kształcenia, zorientowanego na kompetencje społeczne przydatne w rozwiązywaniu problemów. Programowanie zwiększa tutaj zarówno umiejętności logicznego myślenia – program nie będzie działał poprawnie, jeśli źle skonstruujemy jego wewnętrzną logikę – ale i umiejętności pracy w grupie.

Wiele problemów rozwiązać można tylko budując programy według logiki SCRUM, zgodnie z którą mniejsze grupy tworzą elementy większej całości, które potem są w tę całość składane. Pozwala to uczestnikom właśnie na doświadczenie pracy w grupie i szybsze osiągnięcie skomplikowanego rezultatu dzięki równoległemu zaangażowaniu wielu zespołów.

Doświadczenia ruchu otwartościowego pokazują, że udostępnienie kodu pozwala na znacznie skuteczniejsze i szybsze wyłapywanie i poprawianie błędów. W środowisku szkolnym ma to znaczenie nie do przecenienia, gdyż pozwala uczniom na lepsze zrozumienie potencjału jaki daje wspólne działanie online, zarówno w procesie uczenia się, jak i w przyszłym życiu zawodowym.

Przejęcie przez system edukacji wzorców wymiany wiedzy w środowiskach programistów pozwoliłoby uczynić z programowania także lekcję efektywnej współpracy z pomocą sieciowych narzędzi komunikacyjnych. Otwartość kodu eliminuje bowiem bariery geograficzne i czasowe jako przeszkodę we współpracy – grupa uczniów w Szczecinie może pracować wspólnie z uczniami ze szkoły w Przemyślu, albo korzystać z projektów poprzednich roczników, ulepszając je i rozbudowując. Dlatego przynajmniej niektóre programy motywujące do nauki kodowania – konkursy, pro-

jekty oferowane przez organizacje zewnętrzne – powinny promować współpracę między uczniami czy grupami uczniów oraz ponowne wykorzystanie zasobów jako elementy zwiększające kompetencje społeczne w procesie kształcenia.

Rozwiązywanie problemów na przykładzie programowania to także uczenie jakże poszukiwanej w dzisiejszych czasach innowacyjności, na którą polska szkoła obecnie nie bardzo ma pomysł. Warto pamiętać, że otwartość projektów informatycznych wspiera myślenie przedsiębiorcze, ponieważ jej zasadą jest generowanie wartości dodanej na podstawie kodu, który napisał już ktoś inny.

Zatem potraktowanie wyzwania, jakim jest szerokie, systematyczne kształtowanie tych kompetencji, jako przekrojowego zagadnienia leżącego w interesie polskiej szkoły, przyniesie korzyści całemu systemowi edukacji, który zмага się z transformacją od wielu lat. Wyzwanie to adresują również pozostałe rekomendacje. ■

2. Programowanie jako umiejętność powszechna

W oparciu o logikę rozróżnienia kompetencji zaproponowaną przez Komisję Europejską można stwierdzić, że nie każdy użytkownik technologii będzie praktykiem i nie każdy praktyk będzie musiał zarządzać firmą czy budować własny startup. Ponieważ jednak umiejętności związane z programowaniem zyskują na znaczeniu, a celem edukacji jest przygotowanie młodych ludzi do funkcjonowania w rzeczywistości społecznej i zawodowej w przyszłości, tę właśnie przyszłość musimy mieć na systemowym radarze. Celem tworzenia systemów powinno być traktowanie kodowania jako umiejętności powszechnej, obecnej w programie nauczania na różnych etapach.

Jest to istotne ze względu na wspomnianą już lukę w sile roboczej w skali nie tylko Europy, ale i świata. Traktowanie podstaw programowania jako umiejętności powszechnej da każdemu uczniowi możliwość sprawdzenia siebie, swoich zainteresowań i zdolności. Pozwoli także na podjęcie decyzji, jakie znaczenie te umiejętności będą miały w dalszym kształceniu i wyborze ścieżki zawodowej.

Osobnym wyzwaniem jest uwzględnienie w systemie ścieżki dla bardziej zainteresowanych i bardziej uzdolnionych, którzy potencjalnie będą nie tylko użytkownikami technologii ale i zaawansowanymi praktykami. Jest to zupełnie osobne wyzwanie, które w Polsce jest w jakiejś mierze zagospodarowane poprzez konkursy, olimpiady in-

formatyczne, itp. Jednak praca z osobami wykazującymi duże zdolności i zainteresowania jest czym innym niż zapewnienie powszechnego dostępu do nauki programowania oraz powszechnych, podstawowych umiejętności. Oba te zadania można realizować równocześnie, jeśli dobrze funkcjonujący poziom powszechny pozwala na „przesiewowe” wyszukiwanie wybitnie uzdolnionych. ■

3. Dziewczynki i programowanie

Wyzwanie jakim jest szczególne uwzględnienie dziewczynek w komunikowaniu korzyści wynikających z nauki programowania zostało dostrzeżone przez największą europejską inicjatywę pozaszkolną, CoderDojo. Prowadzący program w Irlandii dokładają starań, żeby dziewczynki przychodzące na zajęcia miały wsparcie motywujące je do kontynuacji nauki. Powierzone są im też role liderkie – prowadzenia grup i uczenia mniej zaawansowanych, co zachęca inne dziewczynki do udziału w zajęciach. Większość ośrodków CoderDojo w Irlandii ma parytetową lub bliską parytetową strukturę udziału dziewcząt i chłopców w zajęciach⁸.

Jak pokazuje przykład irlandzki, samo zapisanie w misji przedsięwzięcia chęci uwzględnienia dziewcząt nie wystarcza, konstrukcja programów musi uwzględniać codzienną pracę nad zmianą postaw i podtrzymywanie motywacji. Również otwieranie i wspieranie dążeń liderkich dziewcząt ma znaczenie. Podsumowując, wytyczne ogólne muszą być obecne na poziomie systemo-

wym i obudowane konkretnymi praktycznymi wskazówkami dla wspierania dziewcząt, ich możliwości i aspiracji. ■

4. Infrastruktura jako fundament nauki programowania

Infrastruktura jest jednym z kluczowych narzędzi w tworzeniu systemowych rozwiązań dla programowania. Przez infrastrukturę rozumiemy jej elementy związane z programem nauczania ale i z dostępnym oprogramowaniem i sprzętem.

Raport *Nauka programowania w szkołach – czas na upgrade?* pokazuje, jak ważne jest uwzględnienie nauki programowania od szkoły podstawowej, w sposób dostępny uczniom w tym wieku. Doświadczenia zagraniczne (CoderDojo) pokazują, że jest to możliwe przy użyciu odpowiedniego oprogramowania. Aby zapewnić skuteczność tej edukacji w odniesieniu do wszystkich rodzajów adresatów (przyszłych użytkowników, praktyków oraz liderów), program wszystkich przedmiotów nauczania powinien uwzględniać wykorzystanie nowych technologii i możliwości jakie dają one w przyswajaniu wiedzy, we współpracy, oraz monitorowaniu postępów. Innymi słowy, należy przewidzieć system, w którym celem jest szersze niż obecnie wykorzystywanie komputerów i urządzeń mobilnych w kształceniu.

Należy zwrócić szczególną uwagę na kwestię wykorzystywania przez uczniów własnego sprzętu. Jakkolwiek może to być rozwiązaniem w sytuacji, gdy szkoła nie dysponuje wystarczającym za-

pleczem, to nie powinien być to standard. Sprzęt, do jakiego dostęp mają uczniowie prywatnie jest niejednokrotnie odzwierciedleniem sytuacji życiowej i ekonomicznej rodzin i opiekunów. Należy więc dążyć do stanu, w którym dostępność nowych technologii w procesie uczenia się będzie taka sama dla wszystkich uczniów, bez względu na zasobność majątkową. ■

5. Kompetencje nauczycieli jako katalizator

Bez przekonanych co do ważności tej problematyki oraz dobrze przygotowanych nauczycieli nie będzie dobrej nauki programowania. Przykład pilotażu programu „Mistrzowie kodowania” pokazuje, że to nauczyciele inspirują uczniów do tego rodzaju aktywności, stając się partnerami w rozwiązywaniu programistycznych problemów. Jak zatem doprowadzić do tego, by nauczyciele uczyli się programować i chcieli tę wiedzę przekazywać dalej?

Jednym z rozwiązań jest oczywiście przywrócenie ścieżki kształcenia programistycznego w ramach formalnej edukacji wyższej. To rozwiązanie wymaga współpracy międzyresortowej i międzysektorowej, jak wskazują kolejne rekomendacje.

Inny przykład adresowania braku kompetencji wywodzi się z tradycji otwartościowych oraz modeli uczenia się w systemie *peer-to-peer*. Przykładem systemowego podejścia do wyzwania jakim jest przygotowanie nauczycieli do lepszego

korzystania z technologii jest ROSE (Regionalne Ośrodki Szkoleń e-Learningowych), których założycielką jest dr. Katarzyna Winkowska-Nowak. Stworzyła ona system współpracy między nauczycielami matematyki, zwiększający ich kompetencje i pozwalający na tworzenie wspólnie otwartych materiałów edukacyjnych. Potrzeba rozwoju zawodowego oraz rezultaty współpracy w postaci materiałów edukacyjnych odpowiadających na realne wyzwania w nauczaniu matematyki, motywują nauczycieli do trwałej współpracy z wykorzystaniem platformy e-learningowej⁹. Jest to więc przykład otwartego, wzajemnego edukowania się samych nauczycieli.

Podobne rozwiązanie można zaadoptować dla potrzeb zwiększania kompetencji nauczycieli programowania, nauczycieli innych przedmiotów chcących w większym stopniu wykorzystywać nowe technologie w nauczaniu swoich przedmiotów, a także dla potrzeb wspólnego tworzenia dobrego programu nauczania i materiałów jemu towarzyszących.

Należy też wspierać zabiegi o zwiększenie środków na te cele działaniami kapitalizującymi dotychczasowe osiągnięcia i tworzącymi synergie między już wypracowanymi dobrymi praktykami. Należy również budować partnerstwa, w których nauczanie programowania, a w szczególności przekazywanie nauczycielom odpowiednich kompetencji w tym zakresie jest realizowane w partnerstwie przez instytucje publiczne, biznes i trzeci sektor. ■

6. Szkoła jako katalizator współpracy wielu środowisk

Powyższe rekomendacje pokazują wyraźnie, że jeżeli chcemy wykorzystać potencjał nauki programowania jako pomysłu modernizacyjnego dla polskiej szkoły, realistycznie nie możemy liczyć na to, że eksperyment ten uda się jedynie w oparciu o obecne zasoby kompetencyjne i materialne systemu edukacji. Potrzebna jest współpraca wielu środowisk, by udało się stworzenie systemu uwzględniającego kształcenie na różnych poziomach nauczania i w różnych przedmiotach, zapewniające dostęp do infrastruktury i pomagające wszystkim bez względu na płeć i status majątkowy, nabycie umiejętności odpowiadających zdolnościom i potrzebom.

Szkoła boryka się z przejściem do modelu uwzględniającego dostarczanie nie tylko wiedzy ale i kompetencji społecznych i umożliwiających sterowanie własnym rozwojem po zakończeniu cyklu formalnego kształcenia. Współpraca organizacji, osób i firm, które mogą pomóc w tym przejściu jest nieodzowna.

Nauka programowania jest interesującym przykładem jak firmy zainteresowane zwiększeniem kompetencji przyszłych pracowników, organizacje pracujące w modelu kształcenia nieformalnego oraz inicjatywy aktywizujące nauczycieli mogą stworzyć uczniom możliwości innego podejścia do nauki, nauczycielom nabycie koniecznych kompetencji, a programistom możliwości dzielenia się umiejętnościami w działaniach pozazawodowych.

Rekomendacje – jak wspierać upgrade nauki programowania w szkołach?

Mając na uwadze włączanie dziewcząt w naukę programowania, należy budować partnerstwa z organizacjami już nakierowanymi na takie działania, jak np. Geek Girls Carrots¹⁰ - bowiem mogą one być dobrym akceleratorem budowania komplementarnego systemu, jak i dostarczania konkretnych produktów edukacyjnych. Dostępne publicznie zasoby programu „Mistrzowie kodowania”, takie jak scenariusze lekcji, mogą z kolei uzupełniać materiały finansowane w ramach programu „Cyfrowa szkoła”. Podobnie partnerstwa z firmami w kwestii korzystniejszego nabycia sprzętu i oprogramowania mogą pomóc zagospodarować największe deficyty.

To tylko przykłady ilustrujące pogląd, że szukanie wspólnych celów, synergii i sposobów dzielenia się rezultatami i działaniami są szansą dla lepszej integracji nauczania programowania w systemie kształcenia. Inicjatywy nieformalne i partnerstwa międzysektorowe powinny być traktowane jako kluczowy i niezbędny element systemu kształcenia. Jest je trudniej wystandardyzować, lecz jednocześnie - jako bardziej elastyczne - szybko odpowiadają na realne wyzwania i problemy. ■

7. Strategiczny wymiar nauki programowania w szkołach

Odpowiednie założenia polityki publicznej i właściwe sformułowanie celów strategicznych nie wystarczą do wdrożenia nowoczesnego modelu nauczania programowania w szkołach. Jeśli jednak nauka ta ma być powszechna, a jej poten-

cjał ma mieć efekt transformujący kompetencje dostępne na rynku pracy, niezbędne jest potraktowanie programowania jako priorytetu państwa, wykraczającego poza działania resortu edukacji. Pozwoli to odpowiednio powiązać naukę programowania z priorytetami związanymi z rozwojem kapitału intelektualnego Polski.

Jest to zadanie międzysektorowe, gdyż - jak to już wcześniej wykazaliśmy - stworzenie efektywnego systemu nauczania programowania wymaga współpracy systemu oświaty, systemu szkolnictwa wyższego, ale też na przykład instytucji odpowiedzialnych za rozwój i wspieranie przedsiębiorczości. Oznacza to wpisanie programowania w dokumenty strategiczne, horyzontalną współpracę resortów: edukacji, szkolnictwa wyższego, cyfryzacji oraz gospodarki, ale także włączenie wszystkich zainteresowanych podmiotów w aktywne rzecznictwo programowania jako priorytetu państwa. ■

MISTRZOWIE KODOWANIA – PROGRAMOWANIE JAKO KROK KU NOWOCZESNEJ I OTWARTEJ SZKOLE

Ewaluacja programu *Mistrzowie kodowania* pokazuje wyraźnie, że uczenie programowania w oparciu o idee otwartości, współpracy oraz zniesienie paradygmatu „mistrz-uczeń” przybliży nas do nowoczesnej szkoły. Nowoczesna szkoła uczy rozwiązywania problemów, buduje kompetencje społeczne oraz przygotowuje młodzież do mierzenia się z wyzwaniami rynku pracy. W kontekście debat o tym, czy polska szkoła potrafi uczyć współpracy i budować kapitał społeczny ten potencjał *Mistrzów kodowania* wydaje się nie do przecenienia.

Poprosiliśmy nauczycieli o podzielenie się z nami doświadczeniami z programu oraz ocenę jego mocnych i słabych stron. Przeanalizowaliśmy 52 ankiety ewaluacyjne i 10 wywiadów telefonicznych z nauczycielami. Uczestnicy ewaluacji uczyli w sumie 949 uczniów: 170 w klasach I-III, 162 w klasie IV, 261 w klasie V i 356 w klasie VI.

PROGRAMOWANIE UCZY LOGICZNEGO MYŚLENIA NAKIEROWANEGO NA ROZWIĄZYWANIE PROBLEMÓW

Mistrzowie kodowania zostali ocenieni bardzo wysoko. Nauczyciele uznali program za dobrze przygotowany oraz skutecznie i przyjemnie uczący programowania. Podkreślali, że program uczy „myślenia kreatywnego, strategicznego”, a dzięki niemu uczniowie zdobywają znacznie więcej umiejętności niż tylko programowanie w Scratchu. Uczą się logicznego rozumowania w kategoriach związków przyczynowo-skutkowych, warunków i reguł, ćwiczą cierpliwość, dokładność i współpracę z innymi.

Jedna z nauczycielek, która kiedyś pracowała jako programistka, wspominała, że niesamowite było obserwować, jak szybko i przyjemnie dzieci o niewielkiej jeszcze wiedzy matematycznej

uczyły się kodować. Inni nauczyciele, zwłaszcza ci pracujący w młodszych klasach, podkreślali, że program odwraca obowiązujące do niedawna założenie, że dopiero bardziej rozbudowana i ugruntowana wiedza matematyczna może być bazą do nauki programowania. W pracy ze Scratchem uczniowie napotykali wiedzę w bardziej namacalnej formie konkretnych problemów do rozwiązania i wciągnięci w działanie przyswajali abstrakcyjne pojęcia niemal organicznie.

W praktyce pracy ze Scratchem nauczyciele przekonali się, że jego zastosowanie wykracza daleko poza wąsko pojmowaną lekcję informatyki. Jest on „ciekawym i całościowym narzędziem”, którego można używać do rozwijania innych umiejętności u uczniów. W słowach jednej z nauczycielek, chodzi tu nie tylko o naukę programowania, ale o ogólniejszą i bardziej uniwersalną platformę pracy nad rozwojem wiedzy i umiejętności. Nasi

rozmówcy deklarowali chęć silniejszego włączenia Scratcha w metodykę nauczania.

PROSTOTA I OTWARTOŚĆ NARZĘDZI PROGRAMISTYCZNYCH ZWIĘKSZA ATRAKCYJNOŚĆ NAUKI

Z ankiet i wywiadów przebija przyjemność z używania Scratcha: jest z nim zabawa, jest fajny, „uwiódł” uczniów i nauczycieli, zachwyił. „Zarażili się nim” i zarażają nim innych. Z relacji nauczycieli wyłania się obraz zajęć niezwykle atrakcyjnych dla uczniów, bo wciągających, ale i bardzo satysfakcjonujących dla prowadzących, bo osiągających efekty. Uczniowie kodujący w Scratchu byli „wciągnięci” i „skoncentrowani jak nigdy”, a czas zajęć wydawał się płynąć bardzo szybko i często pojawiały się apele o ich przedłużenie.

Stworzony w MIT Media Lab program jest w tych opowieściach poręcznym narzędziem, które rzadko stawia opór, daje się rozumieć samo przez się, szybko staje się przezroczyste. Relacje o interakcji z oprogramowaniem można odczytywać w kategoriach psychologicznego doświadczenia przepływu (*flow*). W przepływie kompetencje rozwijają się we właściwym balansie między rosnącą trudnością zadania a rosnącymi umiejętnościami wykonującego zadanie, pojawiają się regularne informacje zwrotne korygujące błędy, napędzające motywację i nagradzające sukcesy („udało się!”, „działa!”), a przyjemność płynie z samego trwania aktywności.

PROGRAMOWANIE JEST NARZĘDZIEM W KOMUNIKACJI MIĘDZYPOKOLENIOWEJ

Przyjemności związane z korzystaniem z nowych mediów przez młodych są w kontekście szkoły i domu często obszarem pokoleniowego konfliktu: o wyłączenie telefonu komórkowego w czasie lekcji, o marnowanie czasu w domu na grach komputerowych, o niewłaściwe treści, itd. Przyjemność Scratcha jest tu znacząca dlatego, że raczej łączy a nie dzieli i raczej uskrzydla niż otępia.

Nauczyciele i uczniowie spotykali się w Scratchu nad wspólnym problemem i mówili (programowali) tym samym językiem. Rodzice, na początku zaniepokojeni tym, że dzieci będą „jeszcze dłużej siedzieć przed komputerem”, szybko docenili, że można go używać w inny sposób i pozwalali dzieciom poświęcać dużo czasu na kodowanie. Zdarzało się, że i sami się w nie wciągali. Byli zaskoczeni, widząc, co ich dzieci są w stanie zrobić na komputerze, jeśli przestaną być tylko użytkownikami. Zdaniem nauczycieli oceniających pilotaż jest to jeden z podstawowych efektów programu.

Mistrzowie kodowania proponują nowy sposób myślenia i posługiwania się komputerami, w którym nie chodzi o użytkowanie, lecz o tworzenie. Uczniowie „nauczyli się sterowania komputerem zamiast biernego korzystania z gotowych programów”.

PROGRAMOWANIE ZMIENIA PARADYGMAT „MISTRZ-UCZEŃ” KATALIZUJĄC PRACĘ W GRUPIE

Porozumienie we „wspólnym języku” programowania pozwoliło nauczycielom poprawić kontakt z uczniami i oprzeć go nie na instytucjonalnym autorytecie wiedzy czy podstawy programowej, lecz na „zarażaniu pasją” i wspólnym działaniu. Takie zajęcia są jednak wyzwaniem i dla nauczycieli, i dla uczniów, bo burzą konwencjonalną sytuację pedagogiczną, w której nauczyciel już zna odpowiedź, a uczeń jeszcze nie.

W przypadku Scratcha po pierwsze, nauczyciele wspominali o odwróceniu zwyczajowych ról: „uczyłam się od uczniów, to nowe doświadczenie”. (Antropolodzy współczesności nazywają tę sytuację kulturą prefiguratywną; zmienia się w niej kierunek przekazu niektórych typów wiedzy i wartości). Nasi rozmówcy mówili często o konieczności dotrzymania uczniom kroku, opisywali swoją rolę w niehierarchicznych kategoriach towarzyszenia i współdziałania znacznie chętniej niż w hierarchicznych kategoriach tradycyjnej pedagogiki. Chodziło o „wspólne odkrywanie możliwości Scratcha i wzajemne uczenie się, a także pokazanie uczniom, że uczymy się całe życie i nie wstydzimy się mówić o tym, że czegoś nie wiemy”.

Po drugie, wyzwaniem było to, że uczenie dotyczyło elastycznej umiejętności, a nie kanonicznej wiedzy. Uczniowie dochodzili do rozwiązań „po swojemu”, a dróg rozwiązania było wiele. Rodzi-

to to niepokój — „czy będę umiała pomóc, kiedy uczeń się zapętlę?” — bo w trakcie zajęć regularnie zdarzały się sytuacje nieprzewidziane.

PROGRAMOWANIE PRZEŁAMUJE SCHEMATY MYŚLENIA O KOMPETENCJACH UCZNIÓW

Zajęcia ze Scratchem bywały w szkołach nietypowe, ale właśnie ta nietypowość była ich siłą, bo pozwalała im podważać utarte poglądy i schematy.

Jedna z nauczycielek podkreśla w tym kontekście przełamanie stereotypowego podziału na „humanistów” i „ściślaków”. Okazało się, że „humaniści także mogą kodować”. Innej nauczycielce uczenie kodowania pozwoliło zmienić zdanie o kilku swoich uczniach: „niegrzeczni”, „przeszkadzający” i siedzący w cieniu na matematyce, na zajęciach w ramach *Mistrzów...* pokazali się z zupełnie innej strony, bo praktyczna i wyżej stymulująca forma zajęć lepiej współgrała z ich stylem uczenia się.

PROGRAMOWANIE POMAGA BUDOWAĆ SPOŁECZNOŚCI PRAKTYKI

Dobre oceny efektów programu można przypisać temu, że szkolenia i materiały dla nauczycieli, budzące entuzjazm oprogramowanie („wspólny język”) i angażujące zadania w kolejnych modułach zajęć wytworzyły na zajęciach *Mistrzów kodowania* specyficzne zbiorowości, które antropolog Jean Lave i Etienne Wenger nazywają społecznościami praktyki.

Uczenie się w społecznościach praktyki jest nieodłączne od tworzenia i pielęgnowania więzi społecznych przez wzajemną pomoc i wymianę doświadczeń wobec konkretnych, usytuowanych problemów. Interakcje w ramach wyłaniającej się społeczności praktyki orientują ją na przedsięwzięcie, które jest definiowane jako wspólne, zaś w miarę pracy nad tym przedsięwzięciem rośnie wspólny repertuar zasobów, doświadczeń i wiedzy, do którego członkowie społeczności mogą sięgać, by radzić sobie z kolejnymi zadaniami.

Społeczność wyłania się tu ze wspólnej pracy nad skryptami, wspólnego identyfikowania i usuwania błędów uniemożliwiających ich działanie oraz poczucia, że końcowy wynik jest zbiorowym sukcesem. Relacja jednej z nauczycielek dobrze unaocznia ten mechanizm: „Była taka sytuacja, że uczennica przez dwa zajęcia nie mogła znaleźć błędu, ja też nie mogłam, w końcu pomógł jej inny uczeń i był z tego bardzo dumny”. Ten trop — i słowo „współpraca” — przewija się w wielu wypowiedziach.

PODSUMOWANIE - TRUDNOŚCI NA DRODZE DO SATYSFAKCJI

Charakterystyczne w wielu wypowiedziach jest to, że nauczyciele oceniają program jednocześnie jako przyjemny i satysfakcjonujący, jak i jako trudny i okupiony niepokojami. Często, zwłaszcza wśród tych bez wykształcenia informatycznego, pojawia się obawa: czy sobie poradzę? Czy będę umiał odpowiedzieć na pytania uczniów? Rów-

nież prowadzenie zajęć nie było łatwe: uczniowie rzucali się do komputerów, nie za bardzo chcieli słuchać, co się do nich mówi, próbowali „po swojemu”. „Na pierwszych zajęciach dzieciaki rzuciły się na komputery i niektóre już zaraz były w miejscu, które dla nas miało być ósmym krokiem”.

Ze względu na wcześniejsze doświadczenia z mediami i predyspozycje uczniów w grupach szybko pojawiało się duże zróżnicowanie kompetencji i szybkości pracy. Słabiej radzący sobie uczniowie mogli się zniechęcić, widząc jak dobrze idzie innym. Duże tempo pracy i intensywność wprowadzania nowych umiejętności stawiała uczniów nawet z jedną nieobecnością w trudnej sytuacji.

Tym większe więc było znaczenie rówieśniczej współpracy oraz elastyczności i kreatywności nauczycieli w reagowaniu na wyłaniające się na bieżąco problemy. Wśród nauczycieli ostatecznie te trudności przełożyły się na satysfakcję „poradzenia sobie”, „sprawdzenia się w czymś nowym”, „pokonania wewnętrznych obaw” i na przekonanie, że „można nauczyć się czegoś, co wydaje nam się trudne lub nieosiągalne”.

przypisy

1. Cytowane za Komisją Europejską (empirica, IDC Europe and INSEAD eLab, kwiecień 2013, http://campuses.insead.edu/abu_dhabi/press-releases/2013-e-leadership-study-insead-empirica-idc-europe.cfm)
2. Definicje e-skills stosowane przez Komisję Europejską, http://ec.europa.eu/enterprise/sectors/ict/e-skills/extended/index_en.htm
3. National Curriculum in England: computing programmes of study, <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>
4. Kroes N., Digital for All w „Going Digital - a Parliament Magazine Special Supplement On Women And Technology”, European Parliament 2013, s. 3
5. Komisja Europejska, “EU GDP would be boosted by €9bn if more women worked in ICT”, <http://ec.europa.eu/digital-agenda/en/news/eu-gdp-would-be-boosted-%E2%82%AC9bn-if-more-women-worked-ict>
6. Komisja Europejska, “ICT in schools survey – many children not getting what they need; teachers need more training and support”, http://europa.eu/rapid/press-release_IP-13-341_en.htm
7. Fundacja Nowoczesna Polska, „Cyfrowa przyszłość. Katalog kompetencji medialnych i informacyjnych”, <http://nowoczesna-polska.org.pl/wp-content/uploads/2012/05/Cyfrowa-Przyszlosc-Katalog-Kompetencji-Medialnych-i-Informacyjnych1.pdf>
8. Informacje przekazane przez przedstawiciela organizacji podczas wystąpienia na konferencji Digital Agenda Assembly, Dublin, czerwiec 2013 <https://ec.europa.eu/digital-agenda/en/digital-agenda-assembly-2013>
9. Regionalne Ośrodki Szkoleń e-Learningowych ROSE http://rose.whshost.com/www/index.php?option=com_content&view=article&id=47&Itemid=70
10. <http://geekgirlscarrots.pl>