

Cel, kontekst, opowieści użytkownika

Temat zajęć

Tematem zajęć jest zapoznanie studentów z podstawami budowania "fundamentów" pod projekty informatyczne. W poniższych materiałach zostaną przypomniane wiadomości z wykładu dotyczące definiowania problemów, korzyści i ustalania zakresu aplikacji. Aby skutecznie zacząć definiować wymagania oraz dobrze rozpocząć projekt, potrzebne jest również zidentyfikowanie kontekstu (a więc np. użytkowników i ich ról), a także uzyskanie informacji o potrzebach ludzi chcących skorzystać z naszej aplikacji. Głównym punktem zajęć jest opracowanie przez studentów (w grupach) założeń pomysłów na miniprojekty, przedstawienie ich oraz dyskusja. Dodatkowo, studenci są zobowiązani do przygotowania odpowiedniego dokumentu i podesłania go w terminie wyznaczonym przez prowadzącego.

Postawienie problemu i ustalenie zakresu

Dla każdego projektu informatycznego (i nie tylko), aby miał sens i warto było go realizować, powinien zostać najpierw zdefiniowany **problem**, który powinien zostać rozwiązany lub po prostu motywacja do podjęcia działania. Jest to często bardzo trudna część cyklu tworzenia projektu - czasami problem jest powszechnie znany lub napotykanym bardzo często, jednak należy umieć go dostrzec i odpowiednio zdefiniować. Często zaczyna się realizować konkretny projekt myśląc o tym, "co" chcemy stworzyć, natomiast zapomina się "dlaczego". Jest o tyle ważne, że mając na względzie problem, który chcemy rozwiązać, ma się większą motywację do ukończenia projektu oraz tego, co właściwie chcemy zrobić - jest mniejsza szansa na to, że projekt zacznie ewoluować w zupełnie nieznanym kierunku i nie będzie widać do czego dążymy. Warto sobie również uświadomić znaczenie tego, co chcemy osiągnąć - czy ma to wymiar zaledwie lokalny (jak w przypadku np. aplikacji usprawniającej sesje gier karcianych) czy znacznie większy (jak ułatwienie zarządzania kadrą pracowników na uczelni).

Zdefiniowanie problemu pozwala postawić sobie cel, zwany również **celem biznesowym**. Powodem rozpoczęcia realizacji projektu jest chęć osiągnięcia celu i z tego powodu powinien on być stały (lub względnie stały), mimo często zmieniających się wymagań lub wizji aplikacji. Należy mieć na uwadze, że często na problem napotykają osoby niezwiązane z informatyką, a ich dziedziny mogą być najróżniejsze. Na dodatek, osoby potencjalnie korzystające na powstałej aplikacji mogą mieć różne cele, które mogą być ze sobą sprzeczne.

Postawienie celu determinuje ustalenie **zakresu** funkcjonalności projektu (w bardzo ogólnej postaci). Z uwagi na często bardzo szeroką problematykę, nie zawsze należy zakładać, iż uda się dotknąć każdego aspektu poruszanej tematyki. Dlatego jasne zdefiniowanie zakresu pozwala na skupienie się na konkretnych rzeczach, ułatwia specyfikację wymagań i stanowi też czynnik psychologiczny ("tego nie zrobimy, ale tak zakładaliśmy, czyli nie czujemy, że coś pomijamy").

Przykład problemu: Pod koniec każdego semestru na Politechnice Poznańskiej studenci otrzymują oceny za swoją pracę. Dotyczy to zarówno wykładów, ćwiczeń jak i laboratoriów, a same noty pochodzą z różnych „źródeł” (egzaminów, projektów itd.). Oceny powinny znaleźć się w indeksie oraz na karcie ocen, opatrzone m.in. personaliami wystawiającego oraz jego podpisem, a po zbieraniu wszystkich wpisów student powinien udać się do dziekanatu, w którym następuje weryfikacja i dopuszczenie do następnego semestru (w pozytywnym przypadku). Wiąże się to również z wypełnianiem protokołów przez prowadzących oraz mozolnym zliczaniu punktów ECTS.

Studenci chcieliby uzyskać oceny możliwie jak najszybciej, aby po zbieraniu ich kompletu oddać indeks i kartę ocen do dziekanatu i cieszyć się końcem semestru. Istnieje jednak kilka problemów, które trzeba mieć na uwadze:

- kolejki po wpisy u różnych prowadzących mogą być bardzo długie,
- prowadzący mogą być niedostępni i zwykle niechętni do umawiania się na udzielenie pojedynczego wpisu (a zebranie wystarczającej grupy studentów nie zawsze jest możliwe),
- zarówno indeks jak i (przede wszystkim) karta ocen mogą się zgubić,
- oddawanie indeksów do dziekanatu również wiąże się z długimi kolejkami.

Prowadzący zajęcia również chcieliby jak najszybciej wypełnić formalności. Podobnie jak dla studenta, uzupełnienie indeksów jest bardzo czasochłonne dla pracownika dydaktycznego (szczególnie prowadzącego wykład), podobnie jak przygotowanie protokołu, który następnie trzeba zanieść do dziekanatu. Dodatkowo są określone terminy czasowe na protokoły, o których trzeba pamiętać.

Także pracownicy dziekanatu potrzebowaliby usprawnień. Pod koniec semestru muszą przyjąć wielu studentów oraz wszystkie protokoły od prowadzących. Oceny muszą zostać wpisane do systemu komputerowego oraz zweryfikowane pod kątem punktów ECTS. Nie trzeba mówić, iż jest to bardzo czasochłonne i męczące, a musi zostać zrealizowane w stosunkowo krótkim okresie czasu.

Propozycja rozwiązania: Rozwiązaniem zaistniałego problemu jest przygotowanie systemu informatycznego (o nazwie eProto), który wspomógłby obsługę ocen studentów i znacznie zredukował czas ich wprowadzania i późniejszego przetwarzania. Z uwagi na rozporządzenie Ministerstwa Nauki i Szkolnictwa Wyższego z 2011 roku, możliwe stało się, aby prowadzący wypełniał protokoły elektronicznie, oceny automatycznie trafiały do dziekanatu i były od razu weryfikowane i przygotowywane do zatwierdzenia przez dziekana. W dodatku student mógłby śledzić oceny, które otrzymuje i zgłaszać sytuacje, kiedy wystąpiła jakaś pomyłka.

Zakres projektu:

- zarządzanie protokołami ocen
- śledzenie wpisywanych ocen i możliwość reagowania
- automatyczna weryfikacja ocen

Kontekst

Często samo zdefiniowanie problemu, celu i zakresu, do którego dążymy nie wystarczy - potrzebny jest również **kontekst**, w jakim aplikacja zostanie umieszczona. Można go opisać jako całe środowisko, które istnieje wokół naszej aplikacji i które trzeba mieć na uwadze, aby nie tworzyć projektu oderwanego od rzeczywistości. Na kontekst składają się przede wszystkim użytkownicy, którzy będą korzystać z systemu informatycznego, a konkretnie role, które będą pełnić. Aby dobrze zrozumieć ideę przyświecającą powstawaniu aplikacji, jej znaczenie, a także później wymagania, musimy wiedzieć, dla kogo ją tworzymy (czyli zdefiniować **grupę docelową**) i kto będzie ją wykorzystywał.

Takie role (w pewnym uogólnieniu) nazywamy **aktorami**. Najczęściej program musi spełniać oczekiwania każdej grupy użytkowników i żadnej nie pomijać. Być może istnieją również jakieś zainteresowane strony, którym zależy na tym, aby cel został zrealizowany, ale które nie będą "typowymi" użytkownikami uczestniczącymi w procesie biznesowym (o czym na następnych zajęciach).

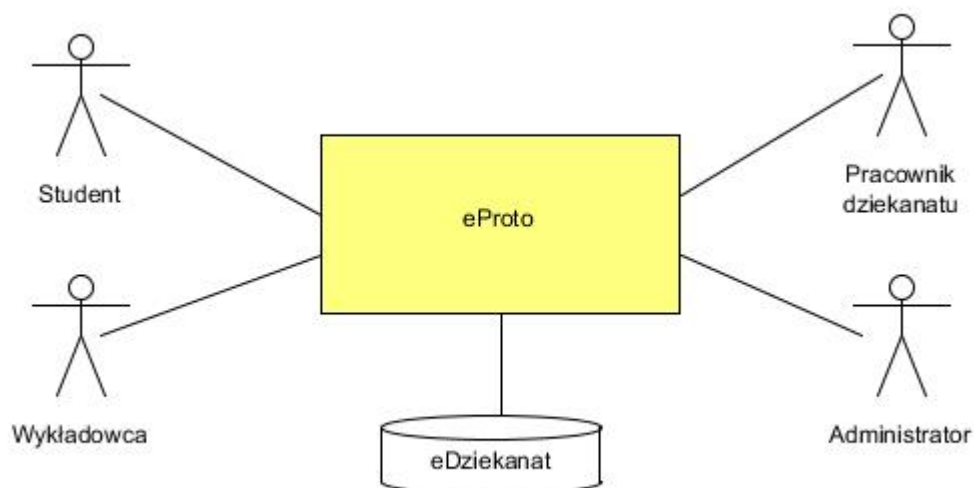
W eProto można wyróżnić następujących aktorów:

- Student
- Prowadzący zajęcia (zwany dla uproszczenia Wykładowcą)
- Pracownik dziekanatu
- Administrator – jest to właśnie przykład „nietypowej” roli, której zadanie jest utrzymywanie systemu i reagowanie na zaistniałe problemy i prośby

Dodatkowo, dla każdego członka grupy docelowej warto opisać korzyści, jakie pojawią się dla niego po wprowadzeniu systemu. W omawianym przykładzie, student będzie mógł bardziej kontrolować swoje oceny i szybciej zgłaszać wątpliwości. Wykładowcy nie będą musieli trudzić się wypełnianiem papierowych formularzy, natomiast pracownicy dziekanatu - nie będą musieli przetwarzać wszystkich danych "ręcznie".

W kontekst aplikacji wchodzić mogą również inne aplikacje lub systemy, które będą wykorzystywane lub same będą pobierać dane. Dobrym przykładem mogą być jakieś portale lub bazy danych udostępniające informacje, które są parsowane i pobierane przez naszą aplikację. Dodatkowo, kontekst obejmuje również wszelkie założenia, które możemy przyjąć (np. istnieją przygotowane serwery po stronie klienta, które możemy wykorzystać) lub zależności, jakie trzeba uwzględnić (np. musimy korzystać z SZBD Oracle).

W przypadku eProto takim systemem, który należy uwzględnić, jest eDziekanat – system dziekanatowy przechowujący i zarządzający wszystkimi informacjami dotyczącymi pracowników i studentów wydziału, który obsługuje listy grup, oceny oraz (nie)zaliczone semestry. Wynikiem opracowania kontekstu jest zwykle **diagram kontekstu** ukazujący w sposób graficzny aktorów oraz ich zależności między sobą a systemem(ami). Przykład dla eProto został przedstawiony na rysunku 1.



Rysunek 1: Diagram kontekstu dla systemu eProto.

Opowieści użytkowników

Wspomniano wcześniej o tym, iż użytkownicy mają różne cele, który chcieliby zrealizować i pewne oczekiwania. Często posiadają również pewną wizję korzystania z systemu, co jest istotne o tyle, że pozwala nam łatwiej zacząć definiować konkretne procesy, wymagania, a nawet prototypować interfejs użytkownika. Takie zdania opisujące wizję są nazywane opowieściami użytkowników (ang. *user stories*).

Opowieści mogą przybierać różną formę. Mogą to być pojedyncze zdania "zapisane" przez różnych użytkowników, czasami na małych karteczkach, które ułatwiają zarządzanie tym, co rzeczywiście powinno zostać zrealizowane.

Jako student chcę mieć możliwość przeglądania ocen, które zostały mi wystawione.

Jako wykładowca chcę mieć dostępną listę studentów, którym mogę wystawić oceny.

Jako pracownik dziekanatu chcę móc w łatwy sposób wydrukować wszystkie protokoły.

Zamiast pojedynczych zdań, opowieści często mogą być bardziej rozbudowane i rzeczywiście przypominać pewną historię oraz opis tego, jak użytkownik chce używać systemu.

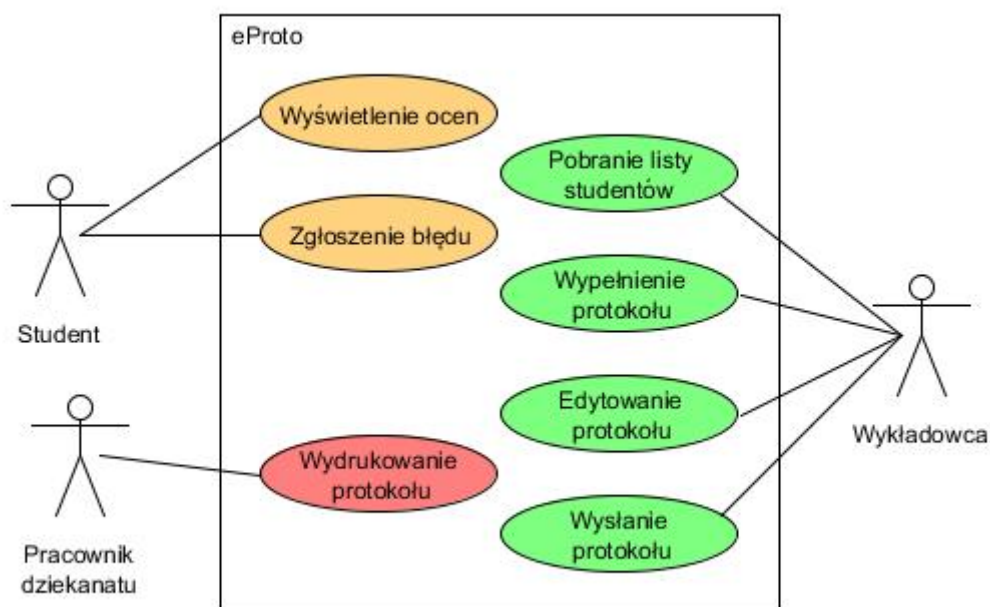
Jestem wykładowcą i po zakończonym egzaminie chcę wystawić oceny swoim studentom. Po określeniu przedmiotu pobieram sobie zatem najpierw listę osób, które uczestniczyły w egzaminie i po jej wyświetleniu wpisuję noty w odpowiednie rubryki. Gdy skończę, mogę łatwo wysłać wypełnione dane do dziekanatu. Może się oczywiście zdarzyć, że gdzieś się pomyłę – w takim wypadku mogę jeszcze raz wyświetlić dany protokół i dokonać poprawki. Ponieważ mam dużo spraw na głowie i mogę łatwo przeoczyć termin wpisywania ocen, system wysłał mi maila przypominającego z pewnym wyprzedzeniem.

Należy pamiętać, że same opowieści nie są traktowane jako wymagania, gdyż czasami są zbyt ogólne i chaotyczne. Są jednak dobrym wstępem do specyfikacji funkcjonalności systemu, a także cech pozafunkcjonalnych.

Diagram przypadków użycia

Jedną z form diagramów UML (które będą omawiane w późniejszej fazie semestru) są diagramy przypadków użycia, które w prosty graficzny sposób przedstawiają funkcjonalność tworzonej aplikacji z uwzględnieniem każdego aktora. De facto, do przygotowania tego diagramu nie jest wymagane stworzenie pełnych przypadków użycia – wystarczą nazwy wymagań.

Przykład dla systemu eProto (przy okrojonej liście wymagań) znajduje się na rysunku 2.



Rysunek 2: Diagram przypadków użycia dla eProto.

Zadanie na zajęcia

Zadanie wynikające z tych zajęć jest dwuetapowe. Pierwszy etap jest w całości wykonywany na zajęciach:

1. Krótkie omówienie problematyki (motywacji) projektu.
2. Krótkie zdefiniowanie celu i zakresu projektu.
3. Określenie grupy docelowej i korzyści wynikających dla jej członków po wprowadzeniu projektu w życie.
4. Wyprecyzowanie aktorów, zależności od innych systemów i założeń.
5. Zapisanie 2-3 opowieści użytkownika i bardzo ogólnego diagramu przypadków użycia.
6. Uzupełnienie przygotowanymi informacjami prezentacji dołączonej do materiałów.

7. Prezentacja (ok. 10-minutowa, w zależności od pozostałego czasu) przed innymi zespołami.

Po każdej prezentacji nastąpi krótka dyskusja o temacie i zaprezentowanych informacjach. Celem debaty jest zwrócenie uwagi na pewne szczegóły, które łatwo ominąć i o których warto pamiętać w przyszłości, a także trening umiejętności zaprezentowania własnego pomysłu.

Drugi etap zadania to rozwinięcie omawianych punktów w postaci dokumentu, którego szablon jest załączony wraz z materiałami. Dokument należy przesłać prowadzącemu tydzień po zakończeniu zajęć (np. jeżeli zajęcia odbyły się w poniedziałek, 3 listopada, to termin przesłania dokumentu upływa w niedzielę, 9 listopada, o 23:59), jeśli nie będzie określone inaczej. Zadanie może zostać ocenione maksymalnie na 10 punktów, a za każdy dzień opóźnienia odejmowane jest 0,5 punkta.