

Nazwa i akronim projektu System Elektronicznego Egzaminowania - eEgzamin	Kierownik projektu Michał Jędrusiak	Opiekun Projektu mgr inż. Michał Wróbel	Numer zlecenia 12@KIOP'10
Zleceniodawca Politechnika Gdańska – Wydział Elektroniki Telekomunikacji i Informatyki, Katedra Inżynierii Oprogramowania		Zleceniobiorca Politechnika Gdańska – Wydział Elektroniki Telekomunikacji i Informatyki, Katedra Inżynierii Oprogramowania	
Nazwa produktu System elektronicznego egzaminowania – Raport końcowy		Data rozpoczęcia prac 02.03.2010r.	Data zakończenia prac 23.04.2010r.
Autorzy oprogramowania: Katarzyna Tyl, Patryk Jar, Michał Jędrusiak, Michał Sokołowski, Michał Wojtowicz			Wersja: 01

1. Zespół projektowy

opiekun: mgr. inż. Michał Wróbel
klient: mgr. inż. Michał Wróbel
uczestnicy: Katarzyna Tyl (KIO)
 Patryk Jar (KAiMS)
 Michał Jędrusiak (KIO)
 Michał Sokołowski (KIO)
 Michał Wojtowicz (KIO)

2. Temat projektu

System elektronicznego egzaminowania - eEgzamin

3. Kontekst projektu

a. Krótka charakterystyka projektu

Przedmiot „Projekt Grupowy” jest realizowany na wydziale Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej na kierunku Informatyka na semestrze 8 oraz 9. W ramach projektu studenci mają za zadanie zrealizować jedno z proponowanych przez uczelnię zadań. Owocem całorocznej pracy ma być produkt oraz specjalistyczna dokumentacja techniczna. Przedmiot ma na celu przygotowanie studentów do pracy w zespole.

Grupa projektowa w składzie: Katarzyna Tyl, Patryk Jar, Michał Jędrusiak, Michał Sokołowski, Michał Wojtowicz wybrała do realizacji wytworzenie systemu elektronicznego egzaminowania. Celem projektu jest stworzenie systemu umożliwiającego przeprowadzanie kolokwii i egzaminów w formie elektronicznej. Docelowy system powinien pozwalać m.in. na definiowanie różnego rodzaju pytań (od testów do pytań otwartych), blokowanie dostępu do wszelkich zasobów elektronicznych komputera, automatycznie sprawdzanie testów. Zalecane jest wykorzystanie technologii OpenSource. Głównym

użytkownikiem systemu będą placówki oświaty a także wszystkie inne instytucje przeprowadzające różnego rodzaju egzaminy pisemne.

b. Cele projektu

Generowanie egzaminu.

Na podstawie pytań wprowadzonych do systemu oraz ustaleniu konfiguracji egzaminu system będzie generował arkusz egzaminacyjny. Podczas przeprowadzenia egzaminu zalecanym byłoby aby pytania dla studentów pojawiały się pojedynczo tak aby ograniczyć możliwość ściągania.

Skrócenie czasu sprawdzania egzaminu.

Egzaminator może automatycznie sprawdzić wyniki pytań testowych oraz pytań zawierających luki. Nie musi już tego wykonywać osobiście.

Przy standardowym sposobie sprawdzania poprawności odpowiedzi udzielonych przez osoby egzaminowane na pytania zamknięte zajmowało dużo czasu gdyż wymagało przejrzenia każdej pracy osobiście. Przy grupie 30 osób trwa to dłużej niż jedną godzinę. Teraz wystarczy tylko wprowadzić odpowiedni plik z odpowiedziami a system automatycznie wykona za nas weryfikację poprawności odpowiedzi pytań zamkniętych. Kilka prac może być sprawdzonych w kilka sekund. Wyniki będą zauważalne gołym okiem. Czas sprawdzenia odpowiedzi otwartych prawdopodobnie pozostanie bez zmian. Egzaminator nie musi już podliczać wszystkich punktów za egzamin. System wykonuje to za niego automatycznie, co też znacznie skraca czas zmęczonemu sprawdzaniem egzaminatorowi.

Zmniejszenie czasu wytwarzania egzaminu.

Osoba mająca potrzebę stworzenia egzaminu może skorzystać z dystrybucji nauczycielskiej systemu. Ma możliwość wprowadzenia całej puli pytań, którą wymyśli o różnych wagach bez znaczenia jaki jest to rodzaj pytania (otwarte, zamknięte, luki, wielokrotnego wyboru). Egzaminator może podać specyfikę egzaminu, ile ma w nim występować pytań danego rodzaju o zadanej trudności.

System sam przygotuje egzamin na podstawie wprowadzonych pytań oraz konfiguracji egzaminu. Egzaminator nie musi już sam mieszać zadań i sprawdzać czy oby każda grupa posiada uczciwy arkusz w którym jest ta sama ilość zadań o takiej samej punktacji i wadze lecz odmiennej treści. Wynik oszczędności czasu widoczny gołym okiem, gdyż zadania egzaminatora kończą się w momencie wprowadzenia zadań i określenia konfiguracji egzaminu.

Pewność studenta o poprawności weryfikacji zadań.

Może zdarzyć się tak, że egzaminator weryfikując poprawność odpowiedzi jest zmęczony i przez pomyłkę nada złą ilość punktów za odpowiedź. System egzaminowania zapewnia nam 100% poprawność weryfikacji pytań testowych oraz pytań typu „luki”. Odpowiedzi są weryfikowane zgodnie z danymi zapisanymi w bazie poprawnych odpowiedzi.

Student wiedząc iż jego odpowiedzi są weryfikowane na podstawie odpowiedzi zawartych w bazie danych ma pewność, że nie nastąpi żadna pomyłka podczas weryfikacji odpowiedzi związana np. ze zmęczeniem osoby sprawdzającej egzamin.

Przeprowadzanie egzaminów drogą elektroniczną.

Egzamin może być przeprowadzony w sieci. Studenci rozpoczynając egzamin dostaną live CD z odpowiednim modulem umożliwiającym rozpoczęcie egzaminu. Prowadzący rozpoczyna egzamin ładując w sieci pytania, które natychmiastowo pojawiają się na monitorach studentów.

Możliwość wielokrotnego wykorzystania egzaminu.

Jeżeli egzaminator zachowa plik z pulą pytań może go wykorzystać kilkakrotnie.

Możliwość automatycznego sprawdzenia poprawności pytań zamkniętych.

Pytanie zamknięte mają zazwyczaj pulę prawidłowych odpowiedzi. System otrzymawszy odpowiedzi studenta może sprawdzić czy dana odpowiedź jest prawidłowa czy też nie.

Zmniejszenie prawdopodobieństwa dokonania oszustwa przez studenta.

Student przystępując do egzaminu musi podać swoje dane. Wszystkie jego odpowiedzi zostają automatycznie zapisane w systemie. Po zakończeniu egzaminu nie ma możliwości dopisywania odpowiedzi oraz podkładania gotowego arkusza odpowiedzi.

c. Charakterystyka klienta

Klientem grupy projektowej jest mgr. inż. Michał Wróbel, Doktorant Politechniki Gdańskiej wydziału Elektroniki Telekomunikacji i Informatyki. Klient stawiał grupie wymagania wytworzone na podstawie obserwacji własnych prac i problemów z jakimi borykają się wykładowcy na uczelni. Klient udostępniał uczestnikom projektu sale uczelniane do badań systemu, uczestnicząc jednocześnie w testach. Wytworzony system jest doskonale przystosowany do warunków jakie dają uczelnie wyższe posiadające dobrze wyposażone sale informatyczne. Klient zwracał jednak uwagę na to aby system był użyteczny dla wszystkich instytucji przeprowadzających egzaminy pisemne.

4. Osiągnięte rezultaty

W pełni funkcjonalny, działający system elektronicznego egzaminowania studentów. System składa się z 3 modułów opisanych poniżej.

Nauczyciel

Aplikacja napisana w Javie, pozwalająca na tworzenie pliku konfiguracji egzaminu.

Potrafi także sprawdzić pytania zamknięte z już przeprowadzonego egzaminu. Dostarcza interfejsu graficznego pozwalającego sprawdzać pytania otwarte, a także posiada opcję eksportu wyników do pliku csv.

Dystrybucja

Dystrybucja Linuksa (Live CD), którą można odpalić na komputerze klasy PC. Pełni rolę serwera WWW oraz DHCP. Pozwala na generowanie indywidualnego egzaminu dla każdego studenta (zgodnego z konfiguracją wytworzoną w Nauczycielu), który jest udostępniany przez serwer WWW jako dokument HTML.

Po zakończonym egzaminie odpowiedzi wszystkich studentów są scalane do jednego pliku, który jest następnie odczytywany i częściowo sprawdzany przez Nauczyciela.

Student

Bardzo okrojona dystrybucja Linuksa (Live CD). Pozwala na uruchomienie systemu operacyjnego na komputerze klasy PC. Po uruchomieniu użytkownik ma do dyspozycji jedynie przeglądarkę internetową, która automatycznie łączy się z serwerem postawionym w module Dystrybucja. Zmiana adresu jest niemożliwa. Wyjście z przeglądarki kończy egzamin.

Metody

Używano zmodyfikowanej metodyki SCRUM. Mimo wykorzystania lekkiej metodyki, z należytą starannością powstała dokumentacja projektu w trakcie jego wytwarzania.

Finalna wersja naszego systemu została przetestowana przez grupę studentów informatyki. Możemy z dumą stwierdzić, że naprawdę działa i jest w pełni gotowy do szerokiego wykorzystania.

Dzięki odpowiedniej dokumentacji i stałej trosce o jakość powstającego kodu jesteśmy pewnie, że projekt może być dalej rozwijany przez kolejny zespół programistów.

5. Proces realizacji projektu

a. Organizacja projektu

Projekt był realizowany przez rok, rozpoczynając 02.03.2010r.. Ostateczny termin rozliczenia się z realizowanych prac to 13.02.2011r.. Podczas 8 oraz 9 semestru nauczania na kierunku Informatyka, wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej. Planowane jest organizowanie spotkań z klientem minimum raz na miesiąc (pomijając okres wakacji).

b. Metodologie

Grupa projektowa korzystała ze zmodyfikowanej metodyki SCRUM. Nie wszystkie spotkania grupy odbywały się osobiście. Uczestnicy mogli obserwować swoje dokonania na Tracku. Przed rozpoczęciem projektu wytworzony został dokument „Product Backlog” na podstawie którego rozdzielane były zadania do kolejnych sprintów. Przy rozliczeniu się z konkretnego sprintu klient wyznaczał funkcjonalności jakie chciały zaobserwować przy kolejnym rozliczeniu. Na podstawie tych wytycznych tworzone były zadania, które rozdzielano pomiędzy uczestników projektu.

c. Wsparcie narzędziowe

Środowisko programistyczne: Netbeans 6.8

Kontener serwletów: Tomcat

Podział pracy: Track z przydzielaniem zadań

Repozytorium:

Katedralny svn - svn używany do umieszczania końcowych efektów każdego sprintu
<https://loqi89.eti.pg.gda.pl/svn/pg/12kiop10/dystrybucja/>

Własny svn - używany do bieżących zmian, zintegrowany z Trac
<http://j0jo.ath.cx/trac/trac-proj/svn>

6. Dokumentacja

a. Techniczna

- Specyfikacja wymagań - SWS_eEgzamin.pdf
- Instrukcje użytkownika – Nauczyciel.pdf, Dystrybucja.pdf, Student.pdf
- Dokumenty Scrum - Product_Backlog.pdf, Sprint_Backlog_1.pdf, Sprint_Backlog_2.pdf, Sprint_Backlog_3.pdf
- Dokument techniczny opisujący proces wytwarzania Systemu Elektronicznego Egzaminowania – eEgzamin.pdf

b. Procesowa

- prezentacje systemu reklamujące system – prezentacja.ppt (stworzona po wykonaniu 2 pierwszych sprintów), prezentacjaKoncowa.ppt (prezentacja promująca, wykonana po zakończeniu prac nad systemem)
- pełny opis spotkań grupy
- szablony dla wszystkich dokumentów technicznych

7. Zmiany w trakcie projektu

a. Organizacja projektu i role członków zespołu

Grupa projektowa miała stały kontakt mailowy, telefoniczny oraz za pomocą internetowych komunikatorów(np.: skype). Wszystkie zmiany w systemie były zapisywane na własnym repozytorium grupowym. Grupa uzyskała również miejsce na serwerze uczelnianym. Repozytorium to było wykorzystywane do zapisu wersji systemu, które zostały przetestowane i zatwierdzone przez całą grupę projektową. Studenci skorzystali z Tracka na którym kierownik grupy rozdzielał zadania. O kontakt z klientem dbał kierownik grupy. Wszystkie spotkania były wcześniej uzgadniane mailowo.

Role członków zespołu przedstawiały się następująco:

Udziałowiec	Opis
Michał Wróbel	klient, odbiorca projektu, zleceniodawca, kierownik projektu Punkt widzenia: obsługa dystrybucji nauczycielskiej, obsługa dystrybucji uczniowskiej, ogólna poprawność działania systemu
Michał Jędrusiak	kierownik grupy, programista, projektant, tester Punkt widzenia: sposób wytworzenia systemu, sposób działania systemu, poprawność działania systemu
Michał Wojtowicz	programista, analityk, projektant, tester Punkt widzenia: sposób wytworzenia systemu, sposób działania systemu, poprawność działania systemu, wymagania wobec systemu, poprawność zebranych wymagań na system,
Michał Sokołowski	programista, projektant, tester Punkt widzenia: sposób wytworzenia systemu, sposób działania systemu, poprawność działania systemu
Katarzyna Tyl	programista, analityk, projektant, tester Punkt widzenia: sposób wytworzenia systemu, sposób działania systemu, poprawność działania systemu, wymagania wobec systemu, poprawność zebranych wymagań na system,

Patryk Jar	programista, projektant, tester Punkt widzenia: sposób wytworzenia systemu, sposób działania systemu, poprawność działania systemu
------------	---

b. Metodologie i narzędzia

Podczas wytwarzania systemu wykorzystano zmodyfikowaną wersję metodyki SCRUM.

Języki programowania zastosowane w projekcie:

- Java:
 - JSP
 - JSLT
- JavaScript:
 - jQuery
- Bash
- HTML, CSS

Środowisko programistyczne: Netbeans 6.8

Kontener serwletów: Tomcat

Podział pracy: Track z przydzielaniem zadań

Repozytorium:

Katedralny svn - svn używany do umieszczania końcowych efektów każdego sprintu
<https://loqi89.eti.pg.gda.pl/svn/pg/12kiop10/dystrybucja/>

Własny svn - używany do bieżących zmian, zintegrowany z Trac
<http://j0jo.ath.cx/trac/trac-proj/svn>

c. Zakres i harmonogram projektu

W trakcie prac nad projektem zrezygnowaliśmy z kilku funkcjonalności zakładanych na początku (np. bardziej rozbudowanego trybu pytań „z lukami”), wyeliminowaliśmy zakładane funkcjonalności wprowadzające tylko problemy w wykorzystaniu platformy (np. możliwość ustawienia by pytanie było jednocześnie dwóch typów) a także dodaliśmy pewne usprawnienia (np. zapis w wynikowym pliku zarówno odpowiedzi studentów, jak i poprawnych odpowiedzi).

d. Rzeczywiste nakłady pracy w stosunku do zakładanych na początku

Dzięki podzieleniu systemu na całkowicie oddzielne, działające niezależnie części mogliśmy zdecydowanie przyspieszyć prace.

Najbardziej czasochłonne okazały się te fragmenty systemu, nad którymi z racji ich rozmiarów musiało pracować kilka osób naraz.

Na samym początku projektu, w fazie analizy i zbierania wymagań określiliśmy, że wszystkie funkcjonalności systemu powinny być możliwe do wytworzenia w ciągu 3 pełnych sprintów + 2 sprintów „roboczych” (jeden w wakacje, drugi pod koniec roku 2010). Sprinty robocze polegały na poprawieniu drobnych błędów

wychwyconych przez klienta, a także na poprawie usability interfejsu graficznego, jakości kodu oraz wykonaniu testów.

8. Podsumowanie

Stworzony przez nas system jest w pełni funkcjonalnym, gotowym do użycia, darmowym produktem. Wachlarz zastosowań Platformy E-Egzamin jest bardzo szeroki i nie zamyka się tylko w murach ETI. Wierzymy, że zarówno w Polsce, jak i na świecie jest wiele osób, które chciałyby skorzystać z potencjału drzemącego w naszym systemie.

Dla nas, młodych programistów, projekt ten był niezwykle okazją do podzielenia się naszymi dotychczasowymi doświadczeniami, jak i zdobycia nowego doświadczenia i umiejętności. Bardzo istotnym okazało się nie tylko wyrównanie środowisk developerskich (do tego celu stworzyliśmy płytę CD z niezbędnym oprogramowaniem, które każdy członek zespołu zainstalował), ale także odpowiednie konwencje programistyczne (choćby wielkość litery, jaką nazywa się projekt w Netbeans ma znaczenie...). W przypadku formatowania kodu przydatne są automatyczne mechanizmy wbudowane w IDE. Z kolei praca w grupie nad kodem bez SVN (lub podobnego) jest niewykonalna. Z pewnością bardzo dobrym rozwiązaniem było też postawienie własnego traca oraz forum. Dzięki temu wiedza i informacje dużo szybciej się propagowały.