

Instytut Informatyki Uniwersytetu Wrocławskiego

Agnieszka Dudek, Piotr Kowalczyk

Dokumentacja projektu *Tablica efektów
kształcenia*

Konstrukcja projektu

Wrocław, 16 stycznia 2019

Wersja 0.6

Tabela 0. Historia zmian dokonywanych w dokumencie

Data	Numer wersji	Opis	Autor
2018-12-19	0.1	Utworzenie dokumentu	Agnieszka Dudek
2019-01-04	0.2	Korekta dokumentu	Agnieszka Dudek
2019-01-08	0.3	Korekta dokumentu	Piotr Kowalczyk
2019-01-09	0.4	Korekta dokumentu	Agnieszka Dudek
2019-01-15	0.5	Korekta dokumentu	Agnieszka Dudek
2019-01-16	0.6	Korekta dokumentu	Piotr Kowalczyk

Spis treści

1. Wprowadzenie	3
1.1. Cel dokumentu	3
2. Testy funkcjonalne dla historyjek użytkownika	3
2.1. Pierwsza historyjka	3
2.2. Druga historyjka	3
2.3. Trzecia historyjka	4
3. Opracowanie pomiarów sprawdzających wymagania niefunkcjonalne zgodnych z normami ISO/IEC 9126 i 5000	4
4. Plan testowania beta	5
5. Plan uwzględniania ryzyka	5
6. Szczegółowe plany	5
6.1. Plan wykonania produktu	5
6.2. Harmonogram i ocena pracochłonności	6
7. Ocena zgodności wykonanych prac z wizją systemu i specyfikacją wymagań	6

1. Wprowadzenie

1.1. Cel dokumentu

Niniejszy dokument ma na celu dokładne przygotowanie testów oraz zaplanowanie zarządzania jakością i ryzykiem programu. Dokument stanowi grudniowe zadanie realizowane w ramach pracowni z inżynierii oprogramowania.

2. Testy funkcjonalne dla historyjek użytkownika

2.1. Pierwsza historyjka

Jako student informatyki, który właśnie układa swój plan zajęć, chcę sprawdzić, ile punktów ECTS z danych grup przedmiotów już zdobyłem. Dzięki temu w kolejnym semestrze będę mógł wybrać takie przedmioty, które uzupełnią punkty brakujące do wymaganego limitu.

Kroki potrzebne do wykonania testu:

- tworzymy profil studenta testowego,
- w zewnętrznym systemie zapisujemy studenta na trzy kursy: *Analiza numeryczna(L)*, *Programowanie obiektowe*, *Przedmiot humanistyczny*, a następnie zaznaczamy, że student zaliczył te przedmioty,
- w zakładce *Tablica efektów* sprawdzamy, czy wszystkie trzy przedmioty są wpisane w odpowiednich kolumnach oraz czy zgadza się liczba punktów ECTS.

2.2. Druga historyjka

Jako studentka informatyki, która zaczyna ostatni semestr studiów, chcę zobaczyć, jakich efektów kształcenia jeszcze nie osiągnęłam i upewnić się, że wszystkie wymagania spełnię w ostatnim półroczu i na czas będę mogła skończyć studia.

Kroki potrzebne do wykonania testu:

- tworzymy profil studenta testowego,
- w zewnętrznym systemie zapisujemy studenta na trzy kursy: *Analiza numeryczna(L)*, *Programowanie obiektowe*, *Przedmiot humanistyczny*, a następnie zaznaczamy, że student zaliczył te przedmioty,
- w zakładce *Tablica efektów* sprawdzamy, czy w przedmiotach niezrealizowanych znajdują się wszystkie wymagane w programie studiów pozostałe przedmioty oraz liczba brakujących punktów ECTS się zgadza.

2.3. Trzecia historyjka

Jako osoba zatrudniona w dziekanacie dowiaduję się, że student wypracował zaliczenie z przedmiotem Algorytmy i struktury danych w lutym, prosi zatem o przedłużenie sesji oraz korektę danych. Chcę, aby po zmianie oceny oraz efektów kształcenia w USOS zaktualizowane zostały dane dotyczące efektów kształcenia w Systemie Zapisów, aby nie było konieczności ręcznego poprawiania danych.

Kroki potrzebne do wykonania testu:

- tworzymy profil studenta testowego,
- w zewnętrznym systemie zapisujemy studenta na kurs *Algorytmy i struktury danych* i zaznaczamy jako przedmiot niezdany,
- w zakładce *Tablica efektów* sprawdzamy, że przedmiot *Algorytmy i struktury danych* jest w kolumnie przedmiotów niezrealizowanych,
- w zewnętrznym systemie zaznaczamy teraz, że kurs *Algorytmy i struktury danych* został zaliczony,
- w zakładce *Tablica efektów* sprawdzamy, że przedmiot *Algorytmy i struktury danych* już jest w kolumnie przedmiotów zrealizowanych.

3. Opracowanie pomiarów sprawdzających wymagania niefunkcjonalne zgodnych z normami ISO/IEC 9126 i 5000

Model jakościowy dostarczony przez standard ISO/IEC 9126-1 definiuje jakość *Tablicy efektów* jako zbiór wielu cech, spośród których największy nacisk kładziemy na płynność działania (wydajność), przejrzystość oraz niezawodność.

W celu przetestowania płynności działania *Tablicy efektów* wyślemy do niej 600 zapytań w ciągu minuty. Odpowiedź powinna trwać nie dłużej niż czterokrotność średniego czasu odpowiedzi *Tablicy efektów* (czas ten zależny jest także od prędkości i obciążenia łącza, zatem musi być mierzony bezpośrednio przed testem).

Przejrzystość interfejsu będzie oceniana przez testerów beta.

W celu przetestowania niezawodności będziemy zbierać informacje dotyczące poszczególnych zapytań. *Tablica efektów* będzie działać poprawnie, jeśli zwróci nie więcej niż jedną niepoprawną odpowiedź na 100 poprawnych zapytań oraz jeśli nie nastąpi więcej niż jeden krytyczny (wymagający ingerencji administratora w celu dalszego działania) błąd na 10000 zapytań (niekoniecznie poprawnych).

4. Plan testowania beta

Przewidujemy dostarczyć projekt na początek semestru zimowego 2019/2020. Aby dobrze przetestować *Tablicę efektów*, udostępniemy program studentom drugiego i trzeciego roku oraz pracownikom dziekanatu na okres wakacyjny. Dzięki temu w razie usterek będziemy mieli czas na naprawienie błędów. Wtedy też studenci nie będą podejmować żadnych ostatecznych decyzji (związanych z głosowaniem, wyborem przedmiotów), zatem będzie to dobry czas na testy.

5. Plan uwzględniania ryzyka

Jak wspomniano w dokumencie *Koncepcja wykonania systemu*, największe ryzyko dotyczące *Tablicy efektów* to ryzyko opóźnień w harmonogramie. Aby temu przeciwdziałać, postaramy się rozłożyć zadania programistyczne i testowe z naciskiem na pierwszą połowę czasu ustalonego w harmonogramie (p. 6.2). Ponieważ ostateczny termin wdrożenia *Tablicy efektów* to połowa września (wtedy zdecydowana większość studentów zacznie interesować się zapisami na kolejny semestr), mamy dodatkowy miesiąc zapasu na ewentualne opóźnienia.

6. Szczegółowe plany

6.1. Plan wykonania produktu

Na początku nawiązany zostanie kontakt z osobami odpowiedzialnymi za rozwój Systemu Zapisów. Po ich akceptacji, omówieniu ich roli w trakcie wykonywania projektu oraz wyznaczeniu programistów do realizacji, rozpocznie się proces wytwarzania oprogramowania. Będzie on zgodny ze standardem używanym w Systemie Zapisów (zawiera m.in. przegląd kodu), uzupełnionym przez dodatkowego testera.

Po wytworzeniu *Tablicy efektów* nastąpią dwie fazy testów. Podczas pierwszej fazy będzie sprawdzona zgodność *Tablicy efektów* z Systemem Zapisów, jego współdziałanie z USOS, a także wydajność i inne wymagania niefunkcjonalne. Po potwierdzeniu poprawności działania *Tablicy efektów* przez ww. testy, rozpocznie się faza testowania beta (p. 4). W trakcie testowania błędy będą poprawiane zaraz po wykryciu.

Po zakończeniu testów osoby pracujące w dziekanacie zostaną przeszkolone do używania *Tablicy efektów*. W tym samym czasie *Tablicy efektów* będzie wdrażana.

6.2. Harmonogram i ocena pracochłonności

Harmonogram jest następujący (wszystkie daty dotyczą roku 2019):

- 1 lutego – 20 lutego — omówienie projektu z osobami odpowiedzialnymi za System Zapisów,
- 21 lutego – 31 maja — wytwarzanie oprogramowania,
- 1 czerwca – 30 czerwca — testy, faza pierwsza,
- 1 lipca – 31 lipca — faza testowania beta,
- 1 sierpnia – 14 sierpnia — szkolenia użytkowników i wdrożenie *Tablicy efektów*.

Niezależnie od powyższego harmonogramu, *Tablica efektów* zacznie być używana przez klientów w połowie września (wtedy studenci zaczynają się interesować ofertą zajęć na kolejny semestr). Zatem czas od 15 sierpnia do 14 września jest dodatkowym buforem, który zostanie wykorzystany w razie opóźnień.

7. Ocena zgodności wykonanych prac z wizją systemu i specyfikacją wymagań

Zgodnie z głównym założeniem dotyczącym *Tablicy efektów* dane wyświetlane studentom muszą być kompletne i przydatne. Będzie to oceniane w trakcie testów beta, kiedy to studenci oraz pracownicy dziekanatu będą zachęceni do wyrażenia swojej opinii.

Wizja wyglądu *Tablicy efektów* nie jest formalnie ustalona. Dopuszczamy wszelkie zmiany, pod warunkiem że będą się one komponować wizualnie z istniejącymi już interfejsami Systemu Zapisów.