

	Concerto – Statut projektu Front systemu sprzedaży biletów na wydarzenia kulturalne	Wersja 1.0
		Data 2012-11-10
Autor	<u>Mateusz Malicki (MM)</u> , Dominik Gebert (DG), Tomasz Cudziło (TC), Mateusz Ochtera (MO)	
Sprawdził	Mateusz Ochtera	
Zatwierdził	Tomasz Cudziło	

1 Statut projektu

1.1 Uzasadnienie projektu

Aktualnie stosowany proces kupowania biletów na wydarzenia kulturalne jest uciążliwy i nie zmienił się znacznie przez ostatnią dekadę. Główną metodą promocji w tym procesie są akcje plakatowe. Możliwe jest uproszczenie i zwiększenie ich efektywności, zwłaszcza wśród głównej grupy docelowej – studentów.

1.2 Cel

Celem projektu *Concerto* jest stworzenie systemu *Software-as-a-Service*. Usługa jest oferowana organizatorom wydarzeń kulturalnych. System *Concerto* pozwala na większe zaangażowanie adresata akcji plakatowej poprzez:

- ułatwienie kupna biletu na wydarzenie od organizatora wydarzenia,
- wygodne przekazywanie zaproszeń na wydarzenie pomiędzy znajomymi,
- szybki sposób dodania wydarzenia do kalendarza osoby bezpośrednio zainteresowanej.

1.3 Oczekiwane rezultaty

Oczekiwane jest nawiązanie współpracy z głównymi organizatorami wydarzeń kulturalnych. Współpraca oznacza przyznanie organizatorowi dostępu do usługi, co bezpośrednio zwiększa jego obroty przy mniejszych nakładach na akcje reklamowe. Z kolei prowizja od sprzedaży biletów dokonanych z polecenia usługi *Concerto*, której wysokość ustalana jest przy nawiązywaniu współpracy, zapewnia pokrycie kosztów utrzymania infrastruktury i zespołu firmy.

1.4 Produkty

System *Concerto* składa się z trzech komponentów: aplikacji mobilnej, serwera obsługi zapytań oraz API dla organizatorów.

1.4.1 Aplikacja mobilna

Aplikacja na urządzenia mobilne kierowana jest do adresatów akcji reklamowych. Umożliwia zakup biletu na reklamowane wydarzenie. Dodatkowo pozwala na

przekazywanie zaproszeń na wydarzenie do znajomych użytkownika. Aplikacja jest cienkim klientem, obróbka zdjęcia i znalezienie adekwatnych informacji jest przekazywane do serwera obsługi zapytań.

1.4.2 Serwer obsługi zapytań

Serwer obsługi zapytań porównuje przesłane przez użytkownika zdjęcie, ze zdjęciami znajdującymi się w bazie danych i zwraca informacje przypisane do plakatu.

1.4.3 API dla organizatorów

API do zarządzania bazą danych wydarzeń jest udostępniane wyłącznie współpracującym organizatorom. Pozwala na zarządzanie informacjami o wydarzeniach. Głównym zadaniem jest dodawanie plakatów wraz z opisem, datą i miejscem nadchodzących wydarzeń.

1.5 Artefakty

1.5.1 Dokumentacja techniczna serwera

Dokumentacja techniczna serwera opisuje wewnętrzną implementację serwera obsługi zapytań. Przeznaczona jest do użytku wewnętrznego właścicieli systemu *Concerto*.

1.5.2 Dokumentacja techniczna API

Dokumentacja techniczna API interfejs programistyczny udostępniany ograniczonemu gronu współpracujących organizatorów.

1.5.3 Dokumentacja procesu testowania

Dokumentacja procesu testowania zawiera przebieg procesów testowania. Do użytku wewnętrznego.

1.5.4 Statut projektu

Statut projektu opisuje najważniejsze zagadnienia projektu. Opisuje cel, założenia i zakres całego projektu.

1.6 Wyznaczniki sukcesu

Projekt musi spełnić szereg wymagań, aby można go było uznać za sukces. Powodzenie projektu zależy od jakości produktu wdrożonego – w głównej mierze nastawienia użytkowników oraz organizatorów.

Oczekujemy że projekt osiągnie następujące rezultaty:

- dla organizatorów – zwiększenie skuteczności akcji reklamowych.
- dla firmy *Concerto* – uzyskanie zysku przy jednym źródle dochodu, tj. od pewnego procentu dochodu ze sprzedaży biletów dokonanych dzięki aplikacji mobilnej.

1.6.1 Wsparcie ze strony organizatorów

Projekt będzie mógł odnieść sukces gdy znaczny procent organizatorów koncertów nawiąże współpracę z właścicielami systemu *Concerto*. Jedynie wtedy możliwe będzie:

- stworzenie odpowiednio dużej bazy informacji o wydarzeniach, że aplikacja mobilna będzie używalna w codziennych warunkach,
- koszt utrzymania infrastruktury serwera obsługującego zapytania nie będzie przekraczał dochodów.

1.6.2 Responsywność systemu

Konieczne jest aby system cechował się wysoką responsywnością. Użytkownik końcowy korzysta z aplikacji mobilnej „w biegu”, operacje powinny zwracać wyniki w możliwie najkrótszym czasie.

1.6.3 Intuicyjność

Wysoka intuicyjność aplikacji mobilnej jest wymagana do uzyskania dużej bazy użytkowników.

1.6.4 Bezawaryjność

Wymagana jest również wysoka bezawaryjność. Aplikacja mobilna nie może mieć negatywnego wpływu na system operacyjny telefonu, sama nie ulegać awariom oraz serwer musi zachować ciągłość pracy. Czynniki niezbędne by utrzymać bazę użytkowników.

1.7 Komunikacja

Wewnętrznie zespół komunikuje się na wielu płaszczyznach, zarówno bezpośrednio jak i za pośrednictwem Internetu. Komunikacja z organizatorami odbywa się w późniejszych fazach rozwoju produktu i organizowana jest przez wyznaczony zespół.

1.7.1 Regularne spotkania

Zespół lub wydzielona grupa spotyka się okresowo w celu omówienia postępów i uzgodnienia oraz koordynacji kolejnych działań.

1.7.2 Notatki ze spotkań

Na każdym oficjalnym spotkaniu sporządzana jest notatka zawierająca poruszone tematy oraz postanowienia.

1.7.3 Grupa dyskusyjna

Do dyspozycji całego zespołu jest oddana również grupa dyskusyjna, gdzie mogą być omawiane bieżące problemy czy wątpliwości, oraz uzgadniane terminy kolejnych spotkań.

1.7.4 Wiki

Każdy moduł projektu posiada stronę wiki przy repozytorium kodu. Wiki zawiera wszystkie informacje projektu omówione na spotkaniach oraz stałe elementy jak protokoły komunikacyjne, zestawienia materiałów o wykorzystywanych technologiach i opisy zasobów.

1.7.5 Protokół komunikacyjny dla wiadomości Git

Wiadomości opisujące zmiany w repozytorium systemu kontroli wersji są wymagane i tworzone przez każdego członka zespołu. Protokół opisuje jak powinna być skonstruowana taka notatka.

1.8 Kontrola & QA

Ważnym aspektem realizacji projektu jest kontrola i zapewnienie jakości. Procesami gwarantującymi uzyskanie wysokiej jakości są tworzenie dokumentacji, testowanie, raportowanie błędów ujawnionych podczas testowania i kontrola postępu. Ze strony każdego programisty oczekuje się, że będzie trzymał się ustanowionej notacji, korzystał ze wzorców projektowych oraz efektywnie wersjonował swój kod.

1.8.1 Tworzenie dokumentacji

Dokumentacji ma na celu stworzenie usystematyzowanego, jednoznacznego i zrozumiałego opisu implementacji każdego elementu systemu.

1.8.2 Testowanie

Testowanie jest ciągłym i równoległym elementem implementacji projektu. Każdy programista stosuje TDD na poziomie testów jednostkowych. Testy integracyjne i akceptacyjne są planowane i tworzone przez programistów odpowiedzialnych za architekturę tego poziomu abstrakcji.

1.8.3 Raportowanie błędów

Obok stosowania TDD wykonywane są regularne testowania kolejnych iteracji systemu. W ich ramach testerzy są zobowiązani do dokumentacji napotkanych awarii oraz działań, które do nich doprowadziły.

1.8.4 Kontrola postępu

Kontrola postępu jest przeprowadzana ciągle w trakcie trwania projektu. Celem jest wczesne wykrycie ewentualnych opóźnień. Pozwala to na przepisanie zasobów do zagrożonych zadań i uzyskanie pewności, że prace zakończą się w wyznaczonym terminie.

1.8.5 Konsekwentna notacja

Cały zespół korzysta z jednej ustanowionej notacji. Pozwala to na utrzymanie spójnej dokumentacji oraz czytelnego kodu.

1.8.6 Wzorce projektowe

Preferowane jest stosowanie znanych wzorców projektowych. Ułatwia to komunikację przy omawianiu logiki implementacji, oraz ułatwia zrozumienie kodu.

1.8.7 Wersjonowanie

Wykorzystanie systemu kontroli wersji jest niezbędne podczas koordynacji i łączenia zmian dokonanych przez wiele osób. Realizowane jest przez rozproszony system kontroli wersji Git.

1.9 Zasoby

Zasoby przyznane zespołowi na czas realizacji projektu.

1.9.1 Zasoby materialne

1.9.1.1 Serwer

Serwer jest ogólnym zasobem świadczącym usługi na rzecz aplikacji mobilnej. W miarę zapotrzebowania i kosztów utrzymania może to być fizyczna maszyna administrowana wewnętrznie, lub wirtualny serwer wynajmowany od firmy zewnętrznej.

1.9.1.2 Urządzenia mobilne

Zespół potrzebuje platformy testowej. Reprezentatywny zespół, aktualnie najpopularniejszych modeli telefonów komórkowych i tabletów. Służą do testowania aplikacji mobilnej.

1.9.1.3 Baza danych wydarzeń kulturalnych

Niezbędna do działania systemu jest oprócz serwera również baza zawierająca informacje o wydarzeniach kulturalnych takie jak ich opis, data, miejsce oraz plik z plakatem.

1.9.1.4 Zasoby czasowe

Zespół ma jeden rok na realizację projektu.

1.9.1.5 Zasoby ludzkie

W zespole każda osoba ma określoną funkcję. Możemy wyróżnić programistów, analityka oraz managera projektu. Niezbędne są również osoby z zewnątrz tj. przedstawiciele współpracujących organizatorów.

1.9.1.5.1 Zespoły programistyczne

Programiści podzieleni są na zespoły pracujące nad aplikacją mobilną, algorytmu pracującego po stronie serwera oraz tworzenia i obsługi serwerów udostępniających API.

1.9.1.5.2 Manager projektu

Osoba koordynująca pracę i odpowiedzialna za prawidłowy przebieg projektu.

1.9.1.5.3 Współpracujący organizatorzy

Firmy, które nawiązały współpracę biznesową z firmą *Concerto*. Biorą czynny udział w tworzeniu bazy wydarzeń kulturalnych. Elementem krytycznym jest by zachować aktualność bazy wydarzeń.

1.9.2 Technologie

Zespół korzysta z różnych technologii w poszczególnych podzespołach projektu. Podstawami są .NET i Ruby on Rails. Dodatkowo wykorzystywany jest LaTeX oraz Git. Dostęp do wszystkich stosowanych technologii jest darmowy, nie ponoszone są dodatkowe koszty związane z uzyskaniem licencji.

1.9.2.1 Ruby

Ruby jest dynamicznym językiem programowania, odznaczający się wysoką ekspresyjnością i wygodą użytkowania. Używany jest m.in. do stworzenia modułu przetwarzającego obrazy.

1.9.2.2 Ruby on Rails

Framework Ruby on Rails jest wykorzystany do stworzenia API dla organizatorów, oraz end-point'u przyjmującego zapytania od aplikacji mobilnych.

1.9.2.3 .NET

Aplikacja mobilna opiera się na .NET. Platforma programistyczna wymagana do stworzenia aplikacji na docelowym systemie operacyjnym Windows Phone 8.

1.9.2.4 LaTeX

Dokumentacja projektu powstaje przy wykorzystaniu LaTeX do składu tekstu.

1.9.2.5 Git

Darmowy, open-source'owy, rozproszony system kontroli wersji.

1.9.3 Szkolenia

Ponieważ zespół nie ma doświadczenia w specyficznej tematyce przetwarzania obrazów wymagane jest przeprowadzanie warsztatów oraz zbudowanie szkoleniowej bazy danych.

1.9.3.1 Warsztaty z przetwarzania obrazu

Szereg intensywnych szkoleń dotyczących przetwarzania obrazów.

1.9.3.2 Przykładowa, kontrolowana baza danych

Dzięki stworzeniu kontrolowanej bazy danych będzie możliwe testowanie powstającego oprogramowania, oraz wykorzystanie jej w czasie szkoleń.

1.10 Ryzyko i propozycje rozwiązań

Projekt może wymagać przyznania większych zasobów lub nie odnieść sukcesu z kilku niezależnych przyczyn. Żadne z opisanych rozwiązań nie wchodzi aktualnie w zakres projektu.

1.10.1 Niska jakość danych od organizatorów

Istnieje możliwość, że organizator będą wysłać niskiej jakości dane, bezużyteczne z punktu widzenia użytkownika aplikacji. W tym przypadku proponowane jest nawiązanie komunikacji z organizatorem w celu wyjaśnienia popełnianych błędów. W przypadku częstego zaniżania jakości publikowanych treści, proponowane jest organizowanie szkoleń dla organizatorów lub stworzenie automatycznego podsystemu oceniającego dane przychodzące z zewnątrz.

1.10.2 Za małe możliwości sprzętowe

Przetwarzanie obrazów jest obciążającym procesem. Platforma sprzętowa może nie być w stanie obsłużyć wielu zapytań jednocześnie i zwrócić odpowiedzi w czasie

spełniającym wymaganie responsywności aplikacji mobilnej. Potencjalnie sytuację można rozwiązać wynajmując odpowiednio wydajniejszy serwer, lub skupić zasoby na optymalizacji istniejącej implementacji serwer obsługi zapytań.

1.10.3 Zły dobór technologii

Platforma Windows Phone może zdobyć marginalny procent rynku smartfonów, co przełoży się na małą liczbę użytkowników. Aby wyeliminować takie ryzyko należy przygotować aplikacje na konkurencyjne mobilne systemy operacyjne iOS i Android.

1.10.4 Brak specjalistycznego doświadczenia zespołu

Zespół nie miał wcześniej do czynienia z przetwarzaniem obrazów. Braki wiedzy członków należy uzupełnić szkoleniem.

1.10.5 Brak zainteresowania produktem końcowym

Posiadacze smartfonów mogą uznać, że oferowana funkcjonalność jest dla nich nieprzydatna. Preferowanymi rozwiązaniami jest akcja promocja, na przykład oferując zniżki na bilety kupione poprzez aplikację mobilną.

1.10.6 Zbyt wysoki koszt utrzymania

Koszt utrzymania serwera może przewyższyć przychody z prowizji od sprzedaży biletów.

2 Historia zmian dokumentu

Lp.	Data	Autor	Temat	Wersja
1.	2012-11-10	MM	Dodano opis elementów mapy pamięci	0.1
2.	2012-11-10	DG	Rozszerzono opisy w sekcjach <i>Ryzyko, Technologie, Kontrola</i>	0.2
3.	2012-11-10	MM	Rozszerzono opisy w sekcjach <i>Materiały, Produkty, Komunikacja</i>	0.3
4.	2012-11-10	DG	Dodano propozycje rozwiązań w sekcji <i>Ryzyka</i>	0.4
4.	2012-11-10	TC	Dodano <i>Uzasadnienie projektu, Cel projektu, Oczekiwane rezultaty</i>	0.5
5.	2012-11-10	MO	Sprawdzono	1.0.rc
6.	2012-11-10	TC	Zatwierdzono	1.0