

Galileo - encyklopedia internetowa

Plan testów

Sławomir Pawlewicz

Alan Pilawa

Joanna Sobczyk

Matek Sobierajski

5 czerwca 2006

Spis treści

1	Wprowadzenie	3
1.1	Cel	3
1.2	Zakres	3
1.3	Definicje	3
1.4	Załączniki	3
2	Przedmioty testów	4
2.1	Testy spójności bazy danych	4
2.2	Testy przypadków użycia	4
2.3	Testy dotyczące interfejsu użytkownika	5
2.4	Testy wydajnościowe	5
2.5	Testy obciążeniowe	5
2.6	Testy wytrzymałościowe	5
2.7	Testy ilościowe	5
2.8	Testy bezpieczeństwa	6
2.9	Testy konfiguracji	6
2.10	Testy instalacji	6
3	Strategie i typy testów	7
3.1	Testy spójności bazy danych	7
3.2	Testy funkcjonalności w oparciu o przypadki użycia	8
3.3	Testy interfejsu użytkownika	8
3.4	Testy wydajnościowe	9
3.5	Testy wytrzymałościowe	9
3.6	Testy ilościowe	10
3.7	Testy bezpieczeństwa i badania dostępu	11
3.8	Testy konfiguracyjne	12
4	Rezultaty	13
4.1	Dzienniki błędów	13
4.2	Raporty	13
5	Zasoby	14
5.1	Zasoby ludzkie	14
5.2	Zasoby sprzętowe	14
6	Historia zmian	15

1 Wprowadzenie

1.1 Cel

Dokument ma za zadanie omówić kwestię przeprowadzania testów jakości tworzonego oprogramowania podczas trwania projektu Galileo. Jako główne cele stawia sobie:

- Wyróżnić te części oprogramowania, które wymagać będą testowania
- Określić, w jaki sposób, z jaką częstotliwością będą przeprowadzane testy
- Wymienić zasoby, jakie będą potrzebne do przeprowadzenia testów

1.2 Zakres

Przeprowadzone testy będą dotyczyć zarówno integralności tworzonego systemu (czyli jego jakości funkcjonalnej), jak i będą badać jego wydajność, bezpieczeństwo oraz skalowalność.

1.3 Definicje

- Galileo nazwa tworzonego projektu
- e-portal.net nazwa firmy, dla której tenże projekt jest tworzony

1.4 Załączniki

- Dokument „Wizja”
- Dokument „Przypadki użycia”
- Dokument „Architektura systemu”

2 Przedmioty testów

Poniżej przedstawiamy listę przedmiotów podlegających testowaniu. Są to przypadki użycia (omówione dokładnie w dokumencie „Przypadki użycia”), wymagania funkcjonalne oraz poza-funkcjonalne.

2.1 Testy spójności bazy danych

Testy polegające na sprawdzeniu, czy można odczytać, usunąć, dodać, zmodyfikować dane zawarte w bazie danych:

- o użytkownikach
- o artykułach
- o pracownikach e-portal.net
- o kategoriach encyklopedii
- o multimediami

2.2 Testy przypadków użycia

- Zarządzanie kategoriami
- Zarządzanie artykułami
- Sporządzenie statystyki
- Zakładanie nowego konta
- Zarządzenie multimediami
- Szukanie informacji w encyklopedii
- Księgowy chce wyliczyć wpływy i wydatki
- Użytkownik ocenia artykuł
- Użytkownik chce zapłacić za korzystanie z encyklopedii
- Użytkownik chce porozmawiać z konsultantem

2.3 Testy dotyczące interfejsu użytkownika

- możliwość wykonania każdego z przypadków użycia
- możliwość uzyskania pomocy
- sprawdzenie przejrzystości interfejsu (ilość kliknięć potrzebnych do wykonania danej operacji)

2.4 Testy wydajnościowe

Testy dotyczą wydajności dostępu przez WWW przy różnych połączeniach z siecią:

- sieć lokalna
- szybkie łącze
- modem

2.5 Testy obciążeniowe

- Weryfikacja szybkości działania bazy danych przy dużym obciążeniu serwera
- Weryfikacja szybkości systemu podczas dużego obciążenia serwera WWW

2.6 Testy wytrzymałościowe

Sprawdzenie, jak zachowuje się system, kiedy baza danych jest zajęta w ponad 85%.

2.7 Testy ilościowe

- Sprawdzenie systemu, gdy dostęp ma do niego 50 użytkowników
- Sprawdzenie systemu, gdy dostęp ma do niego 100 użytkowników
- Sprawdzenie systemu, gdy dostęp ma do niego 200 użytkowników
- Sprawdzenie systemu, gdy dostęp ma do niego 500 użytkowników
- Sprawdzenie systemu, gdy dostęp ma do niego 1000 użytkowników

2.8 Testy bezpieczeństwa

Weryfikacja czynności:

- Tworzenie nowego konta
- Zmiana danych użytkownika
- Usuwanie konta użytkownika

2.9 Testy konfiguracji

Weryfikacja dostępności serwisu przez różne przeglądarki WWW:

- Internet Explorer
- Mozilla
- Netscape Navigator
- Opera
- Konqueror

2.10 Testy instalacji

- Weryfikacja instalacji bazy danych
- Weryfikacja instalacji serwera WWW

3 Strategie i typy testów

W tej części dokumentu przedstawimy dokładniej w jaki sposób będą wyglądały testy poszczególnych części systemu. Omówimy technikę wykonywanych testów, ich cel, potrzebne narzędzia do ich przeprowadzenia oraz kryterium sukcesu.

3.1 Testy spójności bazy danych

Co?	Jak?
Cel testów	Sprawdzenie, jak działa baza danych; czy dostęp do niej nie uszkadza danych oraz czy działanie bazy danych jest niezależne od interfejsu użytkownika.
Technika	<ul style="list-style-type: none">• Testy wymagają od systemu zarządzania bazą danych możliwości wprowadzania lub modyfikacji danych bezpośrednio w bazie danych.• Za pomocą każdej metody dostępu do bazy danych, wprowadzi poprawne i niepoprawne dane.• Za pomocą każdej metody dostępu do bazy danych, zażądać istniejących i nieistniejących danych.• Zbadać, czy dane zostały załadowane w sposób zamierzony.• Zbadać, czy wszystkie spodziewane zdarzenia prawidłowo zaszły w bazie danych.• Upewni się, czy zwrócone dane są poprawne.• Celem ułatwienia dostrzeżenia niespodziewanych zdarzeń, wielkość bazy danych, na której przeprowadza się testy, powinna być możliwie mała.
Potrzebne narzędzia	<ul style="list-style-type: none">• narzędzie do odtwarzania bazowej konfiguracji systemu• narzędzia do monitorowania stanu instalacji• narzędzia administracji bazą danych• narzędzia do generowania danych

Co?	Jak?
Kryteria sukcesu	Wszystkie sposoby dostępu do bazy danych działają zgodnie z założeniem i nie powodują uszkodzenia bazy danych

3.2 Testy funkcjonalności w oparciu o przypadki użycia

Testy będą przygotowane na podstawie skryptów oraz będą sprawdzały działanie poszczególnych przypadków użycia (wszystkie zostały omówione w dokumencie "Przypadki użycia oraz ŚAD"). Testy sprawdzają działanie systemu poprzez interfejs użytkownika - sprawdzana jest reakcja systemu na różne zachowania użytkownika.

Co?	Jak?
Cel testów	Weryfikacja funkcjonalności systemu - pobieranie, modyfikowanie oraz usuwanie danych
Technika	<ul style="list-style-type: none"> • wprowadzenie poprawnych danych i oczekiwanie odpowiednich odpowiedzi • wprowadzenie błędnych danych i oczekiwanie odpowiedniego komunikatu o błędach • sprawdzenie zgodności z założeniami biznesowymi
Potrzebne narzędzia	<ul style="list-style-type: none"> • narzędzie do odtwarzania bazowej konfiguracji systemu • narzędzia do monitorowania stanu instalacji • skrypty do sprawdzania funkcjonalności • narzędzia do generowania danych
Kryteria sukcesu	<ul style="list-style-type: none"> • wykonanie wszystkich testów • ustalenie przyczyn zaistnienia ewentualnych błędów

3.3 Testy interfejsu użytkownika

Testy te będą sprawdzały, czy interakcja między użytkownikiem a systemem przebiega poprawnie.

Co?	Jak?
Cel testów	Weryfikacja, czy użytkownik ma dostęp do wszystkich funkcjonalności systemu oraz czy wymaga on dostatecznie mało czasu
Technika	zbadanie każdego obiektu interfejsu użytkownika
Potrzebne narzędzia	brak
Kryteria sukcesu	wszystkie elementy interfejsu działają poprawnie

3.4 Testy wydajnościowe

Testy obciążeniowe poddają Galileo różnym obciążeniom, aby sprawdzić w nich działanie systemu. Celem jest sprawdzenie działania systemu zwłaszcza podczas obciążenia systemu większego niż maksymalne oczekiwane.

Co?	Jak?
Cel testów	Weryfikacja zachowania systemu podczas: <ul style="list-style-type: none"> • normalnego (oczekiwanego) obciążenia systemu • maksymalnego przewidywanego obciążenia systemu • większego niż ustalone maksymalne obciążenie
Technika	wykorzystanie testów badających działanie przypadków użycia przy większej ilości wykonywanych transakcji
Potrzebne narzędzia	<ul style="list-style-type: none"> • narzędzie ograniczania dostępu do zasobów • narzędzia do monitorowania stanu instalacji • narzędzia do generowania danych
Kryteria sukcesu	wykonanie testów w ustalonym czasie

3.5 Testy wytrzymałościowe

Testy sprawdzają działanie systemu w przypadkach, gdy rywalizacja o zasoby jest duża (znacznie większa niż w normalnych warunkach działania systemu)

Co?	Jak?
Cel testów	<p>Sprawdzenie, jak działa Galileo w przypadku, gdy:</p> <ul style="list-style-type: none"> • wielu użytkowników w tym samym czasie próbuje uzyskać dostęp do tego samego zasobu • jest dostępna niewielka przestrzeń dyskowa • z systemu działa maksymalna ilość użytkowników
Technika	<ul style="list-style-type: none"> • używanie testów opartych na tych, które sprawdzały obciążenie systemu • testowanie przeprowadzone przy obciążonej pamięci • używanie testów przypadków użycia przy wielu klientach działających w tym samym czasie
Potrzebne narzędzia	<ul style="list-style-type: none"> • narzędzie do odtwarzania bazowej konfiguracji systemu • narzędzia do monitorowania stanu instalacji • narzędzie do monitorowania ruchu sieciowego • narzędzia do generowania danych • narzędzie do generowania ruchu sieciowego
Kryteria sukcesu	Przeprowadzone testy, mimo że doprowadziły do ograniczenia systemu, nie doprowadziły do odkrycia defektów oprogramowania.

3.6 Testy ilościowe

Testy te polegają na sprawdzeniu, jak zachowuje się system, kiedy dostarczane są mu duże ilości danych. Są to bardzo ważne testy, ponieważ encyklopedia internetowa będzie musiała obsługiwać i ciągle dodawać nowe dane. Przedmiotem testu jest sprawdzenie, ile danych jest w stanie przetworzyć system w jednostce czasu.

Co?	Jak?
Cel testów	Sprawdzenie, jak działa system, gdy baza danych jest bardzo duża i chce z niej korzystać maksymalna ilość użytkowników.

Co?	Jak?
Technika	<ul style="list-style-type: none"> • Wykorzystanie testów obciążeniowych i wydajnościowych • Należy wypełnić bazę danych • Przy „pełnej” bazie danych należy użyć wielu klientów, którzy będą wykonywać takie same działania przez dłuższy czas.
Potrzebne narzędzia	<ul style="list-style-type: none"> • narzędzia do monitorowania stanu instalacji • narzędzia, które będą ograniczać dostęp do zasobów • narzędzia do generowania danych
Kryteria sukcesu	Przeprowadzone testy, mimo że doprowadziły do ograniczenia systemu, nie doprowadziły do odkrycia defektów oprogramowania.

3.7 Testy bezpieczeństwa i badania dostępu

Testy badają, czy do systemu mają dostęp tylko osoby do tego uprawnione - każdy według przyznanych mu wcześniej uprawnień.

Co?	Jak?
Cel testów	Sprawdzenie, czy osoby mogą się dostać do danych dla nich przeznaczonych (według uprawnień) oraz czy osoby nieuprawnione nie mają do nich dostępu.
Technika	<p>Dla każdego z uprawnień jest sprawdzane:</p> <ul style="list-style-type: none"> • czy jego nadanie umożliwia korzystanie z danych funkcji • czy odebranie go uniemożliwia korzystanie z danych funkcji • czy otrzymanie go nie jest połączone z innymi uprawnieniami
Potrzebne narzędzia	<ul style="list-style-type: none"> • narzędzia do administracji bezpieczeństwem • narzędzia do sondowania systemu
Kryteria sukcesu	Przeprowadzone testy, mimo że doprowadziły do ograniczenia systemu, nie doprowadziły do odkrycia defektów oprogramowania.

3.8 Testy konfiguracyjne

Co?	Jak?
Cel testów	Sprawdzenie, jak działa system dla różnych konfiguracji sprzętowych
Technika	<ul style="list-style-type: none">• Wykorzystanie testów do sprawdzania przypadków użycia• Sprawdzenie, jak działa system, kiedy użytkownik w tym samym czasie korzysta z innych aplikacji
Potrzebne narzędzia	<ul style="list-style-type: none">• narzędzia do monitorowania stanu instalacji• często używane aplikacje użytkownika
Kryteria sukcesu	Oprogramowanie działa poprawnie; obecność innych aplikacji nie ma wpływu na jego działanie

4 Rezultaty

4.1 Dzienniki błędów

Dla każdego testu będą generowane raporty testów, które będą zawierały najważniejsze informacje: id testu, wykryte usterki. Raporty będą w postaci plików tekstowych.

4.2 Raporty

Usterki wykryte w czasie testowania będą rejestrowane przy pomocy Bugzilla.

5 Zasoby

5.1 Zasoby ludzkie

Do przeprowadzania testów będzie potrzebny zespół osób, który będzie generował testy, zajmował się ich wykrywaniem, planowaniem.

Kto?	Liczba	Jaka funkcja?
Kierownik	1	<ul style="list-style-type: none">• planowanie testów• ustalanie obowiązków w zespole• zdobywanie odpowiednich zasobów• ocena przeprowadzonych testów
Analitik testów	1	<ul style="list-style-type: none">• ustalanie szczegółów przeprowadzanych testów• określenie wyników testów• podanie idei testów
Projektant testów	2	ustalanie technicznych szczegółów testów
Tester	4	<ul style="list-style-type: none">• implementacja testów• przeprowadzanie testów• rejestracja wyników testów

5.2 Zasoby sprzętowe

Do przeprowadzenia testów potrzebne będą następujące zasoby:

- Serwer baz danych (2)
- Serwer WWW (1)
- komputery klienckie z różnym połączeniem z internetem (20)
- komputery do tworzenia testów (3)

6 Historia zmian

18.5.06 - pierwsza wersja dokumentu

\$Log: \$