

Projekt BYT

Międzynarodowe zawody sportowe ”Extreme Games”

Autorzy:

Grzegorz Szolc

Tom Szupszyński

Michał Bącal

Hubert Sztajerwald

Michał Ściuba

Andrzej Szuchnicki

Karol Szczepanowski

Spis treści:

1.	Dokument definicji wymagań	3
1.1.	Cel i zakres projektu:	3
1.2.	Wymagania funkcjonalne projektu:	4
1.3.	Wymagania нефункционалне:	7
2.	Plan działań	8
2.1.	Wykres Gantt'a	8
2.2.	Diagram WBS	16
2.3.	Macierz RAM	17
3.	Struktura organizacji zespołu projektowego	19
3.1.	Wydzielenie ról w zespole projektowym	19
3.2.	Struktura organizacji zespołu projektowego	19
4.	Standardy komunikacyjne	20
4.1.	Plan Testowania Oprogramowania	20
4.2.	Plan Zapewnienia Jakości Oprogramowania	22
4.3.	Dokument Specyfikacji Wymagań	24
4.4.	Uwagi ogólne	25
5.	Ankieta dotycząca badania wymagań	26
5.1.	Ankieta	26
5.2.	Ankieta- odpowiedzi udzielone przez klienta	26
6.	Plan zapewnienia jakości	29
6.1.	Cel PZJO (plan zarządzania jakością oprogramowania)	29
6.2.	Zarządzanie	29
6.2.1	Faza wymagań użytkownika i analizy	29
6.2.2	Faza projektu architektury	29
6.2.3	Faza projektowania i konstrukcji	30
6.3.	Przeglądy i audyty	30
6.4.	Kontrola kodu	31
6.5.	Kontrola mediów	31
6.6.	Kontrola dostawców	31
6.7.	Zbieranie, pielęgnacja i utrzymywanie zapisów	31
7.	Plan testowania oprogramowania	32
8.	Oszacowanie złożoności projektu	34
8.1.	Analiza Złożoności Oprogramowania Metodą Zmiennych Funkcyjnych (FPA)	34
8.1.1	Nie skorygowane punkty funkcyjne	34
8.1.2	Korekcja Punktów Funkcyjnych	35
	3. Punkty Funkcyjne (FPs):	35
	4. Plan Szacowania Złożoności Metodą FPA	36
8.2.	Obliczenie punktów funkcyjnych	36

1. Dokument definicji wymagań

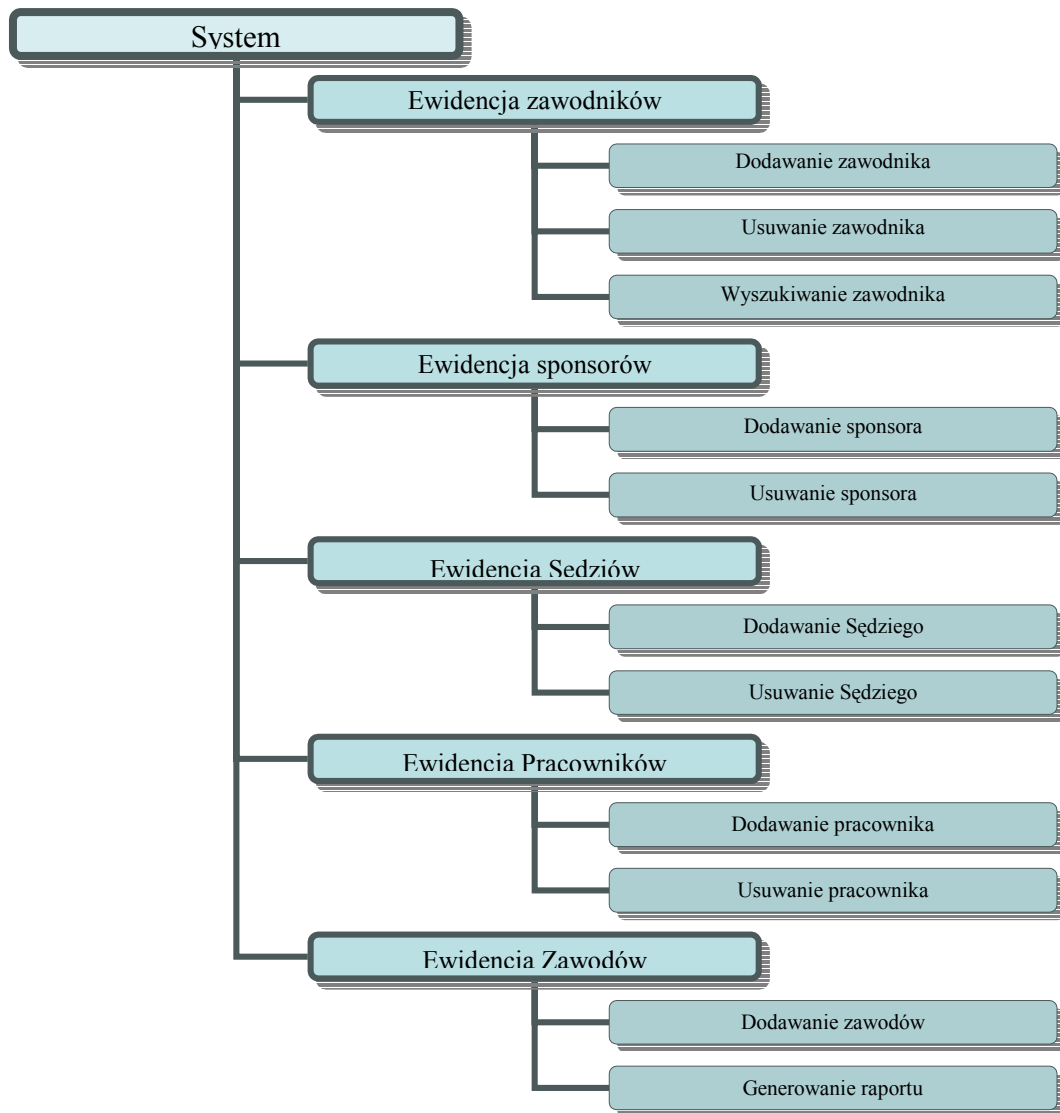
Autor: Grzegorz Szolc

Opis: sformułowanie celu i zakresu projektu, Zdefiniowanie wymagań funkcjonalnych i нефункциональных. Podział wymagań na grupy.

1.1. Cel i zakres projektu:

Celem projektu jest utworzenie systemu informatycznego wspomagającego organizację międzynarodowych zawodów sportowych w zakresie prowadzenia ewidencji zawodników, sędziów, pracowników oraz sponsorów, wspomagania rozplanowywania miejsc i dat zawodów oraz generowania list zwycięzców i raportów po zakończonych zawodach. Systemami zewnętrznymi, z którymi system ma współpracować są sędziowie, pracownicy oraz uczestnicy zawodów.

1.2. Wymagania funkcjonalne projektu:



Nazwa Funkcji	Dodawanie Zawodnika
Opis	Funkcja pozwala na dodawanie zawodnika do bazy
Dane wejściowe	Dane personalna, dyscyplina sportu
Źródło danych wejściowych	Dokumenty oraz informacje dostarczone przez zawodnika
Wynik	Dodanie zawodnika do bazy
Warunek wstępny	Wiek zawodnika ≥ 18 lat, opłacone wpisowe w wysokości 100\$
Warunek końcowy	j.w.
Efekty uboczne	Brak
Powód	Funkcja potrzebna w celu prowadzenia ewidencji zawodników

Nazwa Funkcji	Usuwanie Zawodnika
Opis	Funkcja pozwala na usunięcie zawodnika z bazy
Dane wejściowe	Numer identyfikacyjny lub nazwisko i imię zawodnika
Źródło danych wejściowych	Dane wprowadzone przez pracownika
Wynik	Usunięcie zawodnika Warunek bazy
Warunek wstępny	Zawodnik musi znajdować się w bazie
Warunek końcowy	Brak
Efekty uboczne	Brak
Powód	Funkcja potrzebna w celu prowadzenia ewidencji zawodników

Nazwa Funkcji	Wyszukiwanie Zawodnika
Opis	Funkcja pozwala na wyszukanie i wyświetlenie informacji o zawodniku
Dane wejściowe	Numer identyfikacyjny lub nazwisko i imię zawodnika
Źródło danych wejściowych	Dane wprowadzone przez pracownika
Wynik	Wyświetlenie informacji o zawodniku
Warunek wstępny	Zawodnik musi znajdować się w bazie
Warunek końcowy	Brak
Efekty uboczne	Brak
Powód	Funkcja potrzebna do przeprowadzania zawodów

Nazwa Funkcji	Dodawanie Sponsora
Opis	Funkcja pozwala na dodawanie sponsora do bazy
Dane wejściowe	Dane personalne / dane o firmie
Źródło danych wejściowych	Dokumenty oraz informacje dostarczone przez sponsora
Wynik	Dodanie sponsora do bazy
Warunek wstępny	Brak
Warunek końcowy	Brak
Efekty uboczne	Brak
Powód	Funkcja potrzebna w celu prowadzenia ewidencji sponsorów

Nazwa Funkcji	Usuwanie Sponsora
Opis	Funkcja pozwala na usunięcie sponsora z bazy
Dane wejściowe	Numer identyfikacyjny lub nazwisko / nazwa firmy sponsora
Źródło danych wejściowych	Dane wprowadzone przez pracownika
Wynik	Usunięcie sponsora z bazy
Warunek wstępny	Sponsor musi znajdować się w bazie
Warunek końcowy	Brak
Efekty uboczne	Brak
Powód	Funkcja potrzebna Warunek celu prowadzenia ewidencji sponsorów

Nazwa Funkcji	Dodawanie Sędziego
Opis	Funkcja pozwala na dodawanie sędziego do bazy
Dane wejściowe	Dane personalne
Źródło danych wejściowych	Dokumenty oraz informacje dostarczone przez sędziego
Wynik	Dodanie sędziego do bazy
Warunek wstępny	Sędzia musi posiadać uprawnienia sędziowskie
Warunek końcowy	j.w.
Efekty uboczne	Brak
Powód	Funkcja potrzebna w celu prowadzenia ewidencji sędziów

Nazwa Funkcji	Usuwanie Sędziego
Opis	Funkcja pozwala na usunięcie sędziego z bazy
Dane wejściowe	Numer identyfikacyjny lub nazwisko sędziego
Źródło danych wejściowych	Dane wprowadzone przez pracownika
Wynik	Usunięcie sędziego z bazy
Warunek wstępny	Sędzia musi znajdować się w bazie
Warunek końcowy	Brak
Efekty uboczne	Brak
Powód	Funkcja potrzebna w celu prowadzenia ewidencji sędziów

Nazwa Funkcji	Dodawanie Pracownika
Opis	Funkcja pozwala na dodawanie pracownika do bazy
Dane wejściowe	Dane personalne
Źródło danych wejściowych	Dokumenty oraz informacje dostarczone przez pracownika
Wynik	Dodanie pracownika do bazy
Warunek wstępny	Brak
Warunek końcowy	Brak
Efekty uboczne	Brak
Powód	Funkcja potrzebna w celu prowadzenia ewidencji pracowników

Nazwa Funkcji	Usuwanie Pracownika
Opis	Funkcja pozwala na usunięcie pracownika z bazy
Dane wejściowe	Numer identyfikacyjny lub nazwisko pracownika
Źródło danych wejściowych	Dane wprowadzone przez pracownika
Wynik	Usunięcie pracownika z bazy
Warunek wstępny	Pracownik musi znajdować się w bazie
Warunek końcowy	Brak
Efekty uboczne	Brak
Powód	Funkcja potrzebna w celu prowadzenia ewidencji pracowników

Nazwa Funkcji	Dodawanie Zawodów
Opis	Funkcja pozwala na dodanie zawodów do bazy
Dane wejściowe	Miejsce oraz data zawodów, dyscyplina sportu, skład sędziowski, sponsorzy, patronat, obowiązujący język
Źródło danych wejściowych	Dane wprowadzone przez pracownika oraz wygenerowane automatycznie przez system
Wynik	Dodanie zawodów do bazy
Warunek wstępny	Wystarczająca ilość zawodników oraz sędziów
Warunek końcowy	Brak
Efekty uboczne	Brak
Powód	Dodaje zawody do bazy, co ułatwia ich organizację

Nazwa Funkcji	Generowanie Raportu
Opis	Funkcja generująca raport po zawodach sportowych
Dane wejściowe	Wyniki uzyskane przez zawodników oraz inne dane wprowadzone przez sędziów
Źródło danych wejściowych	Dane zgromadzone w systemie
Wynik	Wygenerowanie raportu
Warunek wstępny	Zawody zostały zakończone, wszystkie niezbędne dane zostały wprowadzone do systemu
Warunek końcowy	j.w.
Efekty uboczne	Brak
Powód	Funkcja niezbędna w celu generowania raportów po zawodach sportowych

1.3. Wymagania niefunkcjonalne:

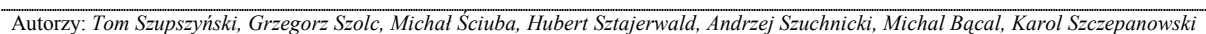
Wymaganie	Motywacja
Raport powinien się generować w maksymalnie 5 sekund	Wygoda użytkownika, wymagania rynku
Liczba rekordów w bazie danych powinna być nie mniejsza niż 10000	W przeciwnym wypadku może dojść do sytuacji, gdzie nie będzie już miejsca na dodawanie nowych pozycji
Czytelny, intuicyjny interfejs użytkownika	Łatwość użytkowania, krótki czas szkolenia pracowników
System powinien być niezawodny oraz umożliwiać tworzenie kopii zapasowych lub też tworzyć je automatycznie	Bezpieczeństwo danych
System powinien być zabezpieczony przed dostępem niepowołanych osób	Bezpieczeństwo danych
System powinien być zgodny z systemami Windows	Najpopularniejsze środowisko, brak konieczności instalowania innych systemów operacyjnych na dostępnych komputerach

2. Plan działań

Autor: Karol Szczepanowski

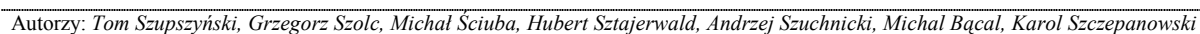
Opis: Szczegółowe określenie planu działań w projekcie oraz kontrola postępów prac.
Zastosowanie diagramów WBS, macierzy RAM oraz wykresu Gantt.

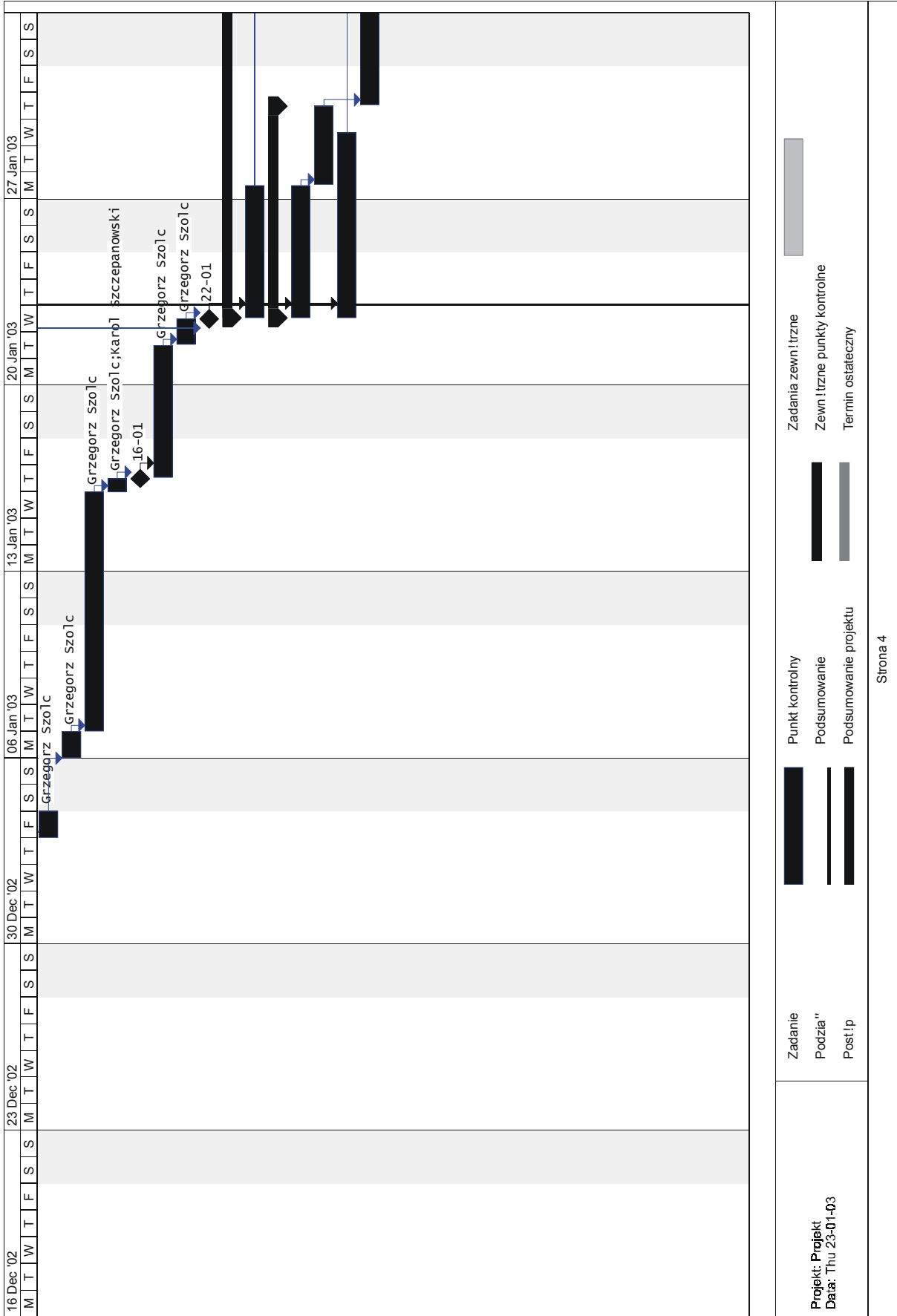
2.1. Wykres Gantt

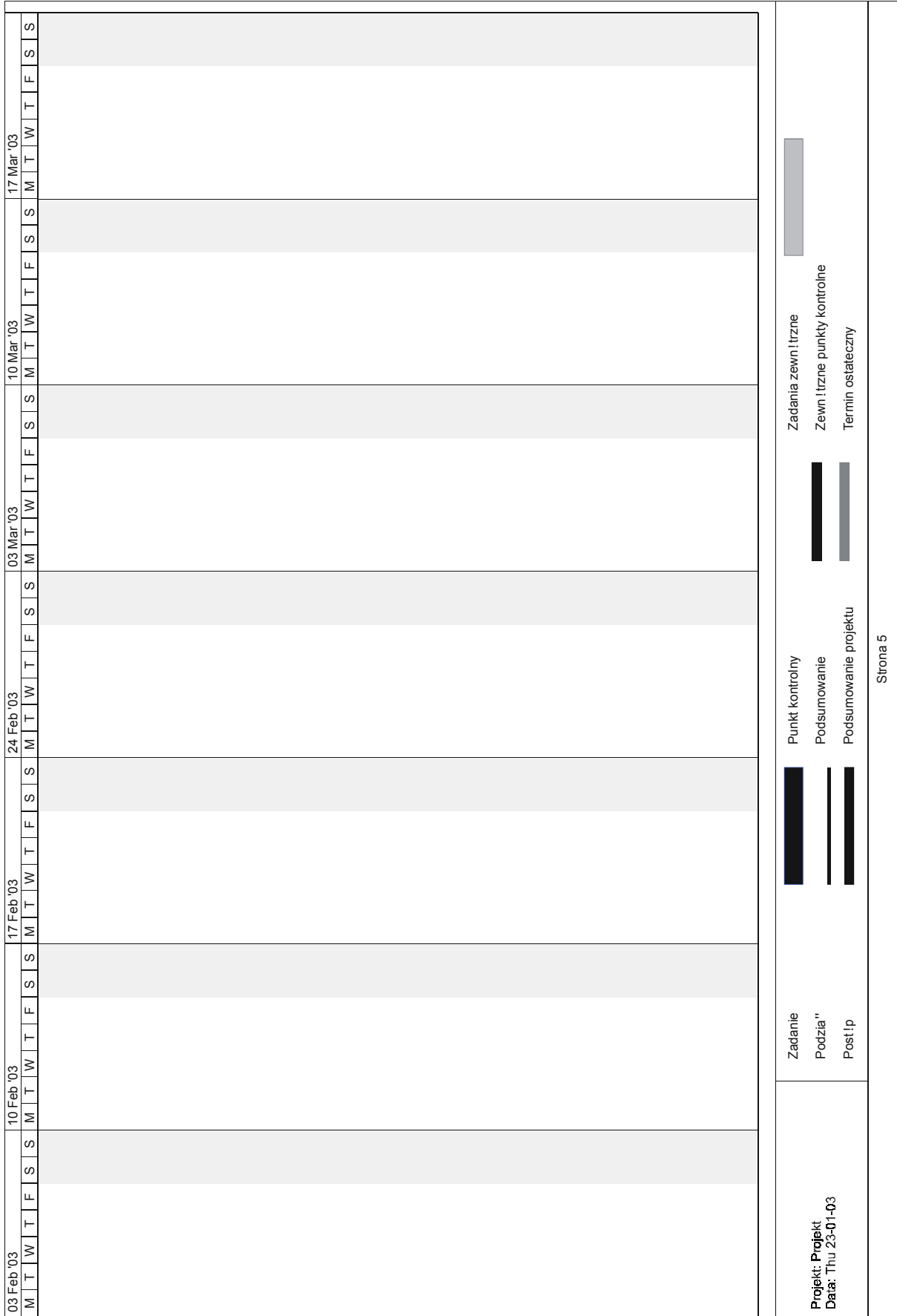


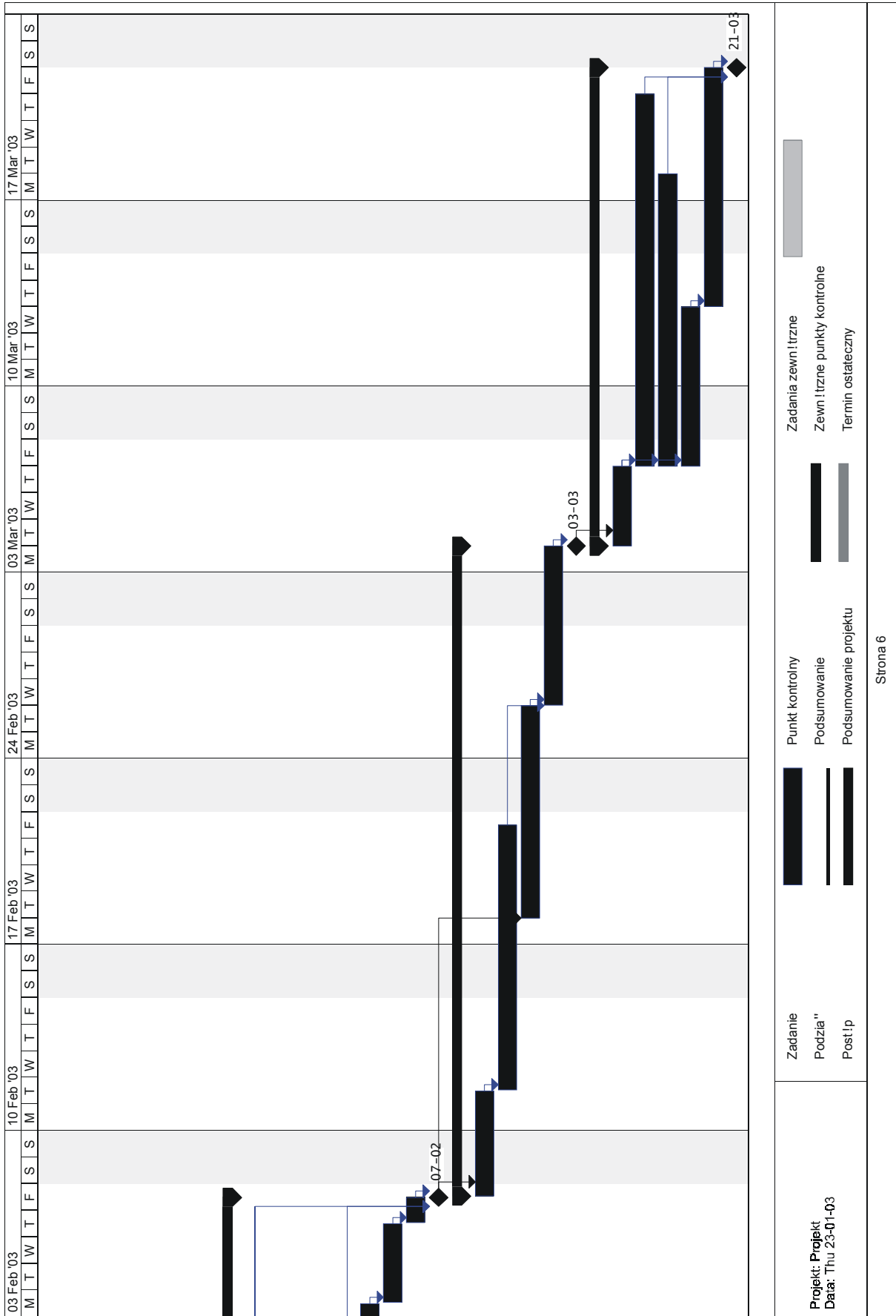


ID		Nazwa zadania	Duration	Resource Names	02 Dec '02							09 Dec '02																				
					S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S												
32		Identyfikacja klas i obiektów	1 day	Grzegorz Szolc																												
33		Weryfikacja zidentyfikowanych klas	1 day	Grzegorz Szolc																												
34		Budowa diagramu klas	7 days	Grzegorz Szolc																												
35		Badanie poprawności modelu klas	0.5 days	Grzegorz Szolc; Karol Szczepanowski																												
36		Przyjęcie modelu klas	0 days																													
37		Praca nad scenariuszmi trudniejszych funkcji systemu	3 days	Grzegorz Szolc																												
38		Ocena wytworzonego modelu	1 day	Grzegorz Szolc																												
39		Koniec fazy analizy	0 days																													
40		Faza projektowania	12 days																													
41		Identyfikacja funkcji trudnych do zrealizowania	3 days																													
42		Uzyskanie danych otrzymanych z fazy analizy	6 days																													
43		Identyfikacja funkcji niezwiązanych z dziedziczeniem projektów	3 days																													
44		Uszczegóławianie modelu klas	3 days																													
45		Praca nad GUI	5 days																													
46		Przeniesienie modelu analitycznego na model relacyjny	2 days																													
47		Praca nad optymalizacją projektu	3 days																													
48		Ocena modelu projektowego	1 day																													
49		Przyjęcie modelu projektowego	0 days																													
50		Faza implementacji	16, 5 days																													
51		Opracowanie dokumentu o standardach programowania	2 days																													
52		Implementacja	8 days																													
53		Testowanie na bieżąco fragmentów kodu	6 days																													
54		Szczegółowe przetestowanie całego systemu	4 days																													
55		Koniec fazy implementacji	0 days																													
56		Faza instalacji	14 days																													
57		Zainstalowanie systemu u zleceniodawcy	3 days																													
58		Przeprowadzenie szkoleń po stronie Klienta	10 days																													
59		Wypełnianie baz danych	7 days																													
60		Przeprowadzenie ostatecznych testów	4 days																													
61		Ostateczne poprawienie błędów	7 days																													
62		Zatwierdzenie systemu przez zleceniodawcę	0 days																													
Projekt: Projekt Data: Thu 23-01-03					Zadanie		Punkt kontrolny		Zadania zewnętrzne		Zewnętrzne punkty kontrolne		Termin ostateczny																			
					Podział		Podsumowanie		Podsumowanie projektu																							
					Postęp																											
					Strona 2																											









2.2. Diagram WBS

Ze względów technicznych poziomy w diagramie zostały zaprezentowane przy użyciu indeksów.

WBS									
ID	WBS	Task Name	Duration	Start	Finish	Predecessors	Resource Names		
1	1	Faza strategiczna	12 days	Mon 02-12-02	Tue 17-12-02				
2	1.1	Stworzenie krótkiego opisu produktu	1 day	Mon 02-12-02	Mon 02-12-02		Grzegorz Szolc		
3	1.2	Zebrać informacje wymaganych do podjęcia decyzji o rozpoczęciu produktu	5 days	Tue 03-12-02	Mon 09-12-02	2	Grzegorz Szolc		
4	1.3	Powołanie zespołu	2 days	Tue 03-12-02	Wed 04-12-02	2	Karol Szczepanowski; Tom Szupczyński		
5	1.4	Wybór narzędzi projektowych	4 days	Tue 03-12-02	Fri 06-12-02	2	Karol Szczepanowski; Grzegorz Szolc		
6	1.5	Analiza możliwości technicznych	2 days	Mon 09-12-02	Tue 10-12-02	5	Michał & ciuba		
7	1.6	Oszacowanie kosztów projektu	1 day	Wed 11-12-02	Wed 11-12-02	6,4,3	Karol Szczepanowski; Michał & ciuba		
8	1.7	Opracowanie analizy ryzyka	2 days	Thu 12-12-02	Fri 13-12-02	7	Karol Szczepanowski		
9	1.8	Podjęcie decyzji o rozpoczęciu projektu	0 days	Fri 13-12-02	Fri 13-12-02	8	Karol Szczepanowski		
10	1.9	Przydział stanowisk w zespole i określenie zakresu odpowiedzialności poszczególnych osób	1 day	Mon 16-12-02	Mon 16-12-02	9	Tom Szupczyński		
11	1.10	Stworzenie standardów komunikacyjnych	2 days	Mon 16-12-02	Tue 17-12-02	9	Michał & ciuba		
12	1.11	Stworzenie standardów komunikacyjnych	0 days	Tue 17-12-02	Tue 17-12-02	11;10			
13	2	Faza analityczna	25,5 days	Wed 18-12-02	Wed 22-01-03				
14	2.1	Opracowanie planu testów	2 days	Wed 18-12-02	Thu 19-12-02	12	Andrzej Szuchnicki		
15	2.2	Opracowanie planu zapewnienia jakości	2 days	Wed 18-12-02	Thu 19-12-02	12	Michał & ciuba		
16	2.3	Analiza otrzymanych wymagań	3 days	Wed 18-12-02	Fri 20-12-02	12	Grzegorz Szolc		
17	2.4	Ankieta	5 days	Mon 23-12-02	Fri 27-12-02				
18	2.4.1	Przygotowanie ankiety precyzyjnej wymagań	3 days	Mon 23-12-02	Wed 25-12-02	16	Hubert Sztajenwald		
19	2.4.2	Przeprowadzenie ankiety	1 day	Thu 26-12-02	Thu 26-12-02	18	Hubert Sztajenwald		
20	2.4.3	Analiza danych i przygotowanie wniosków z ankiety	1 day	Fri 27-12-02	Fri 27-12-02	19	Hubert Sztajenwald; Grzegorz Szolc		
21	2.5	Konsultacje ze zleceniodawcą	1 day	Mon 23-12-02	Mon 23-12-02	16	Grzegorz Szolc		
22	2.6	Uściślenie wymagań	3 days	Mon 30-12-02	Wed 01-01-03	21;20	Grzegorz Szolc		
23	2.7	Stworzenie dokumentu definicji wymagań	1 day	Thu 02-01-03	Thu 02-01-03	22	Grzegorz Szolc		
24	2.8	Przyjęcie ostatecznej wersji wymagań	0 days	Thu 02-01-03	Thu 02-01-03	23			
25	2.9	Grupowanie wymagań użytkownika	1 day	Fri 03-01-03	Fri 03-01-03	24	Grzegorz Szolc		
26	2.10	Praca nad modelem przypadków użycia	8 days	Mon 06-01-03	Wed 13-01-03				
27	2.10.1	Opracowanie ogólnego diagramu use-case	4 days	Mon 06-01-03	Thu 09-01-03	25	Grzegorz Szolc		
28	2.10.2	Przedyskutowanie diagramu przypadków użycia	2 days	Fri 10-01-03	Mon 13-01-03	27	Grzegorz Szolc		
29	2.10.3	Uściślenie diagramu przypadków użycia	2 days	Tue 14-01-03	Wed 15-01-03	28	Grzegorz Szolc; Karol Szczepanowski		
30	2.10.4	Przyjęcie modelu przypadków użycia	0 days	Wed 15-01-03	Wed 15-01-03	29			
31	2.11	Praca nad modelem klas	9,5 days	Fri 03-01-03	Thu 16-01-03				
32	2.11.1	Identyfikacja klas i obiektów	1 day	Fri 03-01-03	Fri 03-01-03	23	Grzegorz Szolc		
33	2.11.2	Weryfikacja zidentyfikowanych klas	1 day	Mon 06-01-03	Mon 06-01-03	32	Grzegorz Szolc		
34	2.11.3	Budowa diagramu klas	7 days	Tue 07-01-03	Wed 15-01-03	33	Grzegorz Szolc		
35	2.11.4	Badanie poprawności modelu klas	0,5 days	Thu 16-01-03	Thu 16-01-03	34	Grzegorz Szolc; Karol Szczepanowski		
36	2.11.5	Przebieg modelu klas	0 days	Thu 16-01-03	Thu 16-01-03	35			
37	2.12	Praca nad scenariuszami trudniejszych funkcji systemu	3 days	Thu 16-01-03	Tue 21-01-03	36	Grzegorz Szolc		
38	2.13	Ocena wytworzonego modelu	1 day	Tue 21-01-03	Wed 22-01-03	37	Grzegorz Szolc		
39	2.14	Koniec fazy analizy	0 days	Wed 22-01-03	Wed 22-01-03	14;15;38			
40	3	Faza projektowania	12 days	Wed 22-01-03	Fri 07-02-03				
41	3.1	Identyfikacja funkcji trudnych do zrealizowania	3 days	Wed 22-01-03	Mon 27-01-03	39			
42	3.2	Uczelnianie danych otrzymanych z fazy analitycznej	6 days	Wed 22-01-03	Thu 30-01-03				
43	3.2.1	Identyfikacja funkcji nie związanych z dziedziną projektów	3 days	Wed 22-01-03	Mon 27-01-03	39			
44	3.2.2	Uściślenie modelu klas	3 days	Mon 27-01-03	Thu 30-01-03	43			
45	3.3	Praca nad GUI	5 days	Wed 22-01-03	Wed 29-01-03	39			
46	3.4	Przeniesienie modelu analitycznego na model relacyjny	2 days	Thu 30-01-03	Mon 03-02-03	44			
47	3.5	Praca nad optymalizacją projektu	2 days	Mon 03-02-03	Thu 06-02-03	46			
48	3.6	Ocena modelu projektowego	3 days	Mon 03-02-03	Fri 07-02-03	47			
49	3.7	Przyjęcie modelu projektowego	1 day	Thu 06-02-03	Fri 07-02-03	48;41;45			
50	4	Faza implementacji	16,5 days	Fri 07-02-03	Mon 03-03-03				
51	4.1	Opracowanie dokumentu o standardach programowania	2 days	Fri 07-02-03	Tue 11-02-03	49			
52	4.2	Implementacja	8 days	Tue 11-02-03	Fri 21-02-03	51			
53	4.3	Testowanie na biegu co fragmentów kodu	6 days	Tue 18-02-03	Tue 25-02-03	49			
54	4.4	Szczegółowe przetestowanie całego systemu	4 days	Wed 26-02-03	Mon 03-03-03	53;52			
55	4.5	Koniec fazy implementacji	0 days	Mon 03-03-03	Mon 03-03-03	54			
56	5	Faza instalacji	14 days	Tue 04-03-03	Fri 21-03-03				
57	5.1	Zainstalowanie systemu u zleceniodawcy	3 days	Tue 04-03-03	Thu 06-03-03	55			
58	5.2	Przeprowadzenie szkoleń po stronie klienta	10 days	Fri 07-03-03	Thu 20-03-03	57			
59	5.3	Wypełnianie baz danych	7 days	Fri 07-03-03	Mon 17-03-03	57			
60	5.4	Przeprowadzenie ostatecznych testów	4 days	Fri 07-03-03	Wed 12-03-03	57			
61	5.5	Ostateczne poprawienie bazy danych	7 days	Thu 13-03-03	Fri 21-03-03	60			
62	5.6	Zatwierdzenie systemu przez zleceniodawcę	0 days	Fri 21-03-03	Fri 21-03-03	58;59;61			

2.3. Maciej RAM

		Grzegorz Szolc	Tom Szupczyński	Michał Bącal	Hubert Sztajerwald	Michał Ściuba	Andrzej Szuchnicki	Karol Szczepanowski
1	Faza strategiczna							
1.1	Stworzenie krótkiego opisu produktu	X						
1.2	Zebranie informacji wymaganych do podjęcia decyzji o rozpoczęciu produktu	X						
1.3	Powołanie zespołu		X					X
1.4	Wybór narzędzi projektowych	X						X
1.5	Analiza możliwości technicznych					X		
1.6	Oszacowanie złożoności projektu			X				X
1.7	Opracowanie analizy ryzyka			X				X
1.8	Podjęcie decyzji o rozpoczęciu projektu							X
1.9	Przydział stanowisk w zespole i określenie zakresu odpowiedzialności poszczególnych osób		X					
1.10	Stworzenie standartów komunikacyjnych			X				
2	Faza analizy							
2.1	Opracowanie planu testów						X	
2.2	Opracowanie planu zapewnienia jakości					X		
2.3	Analiza otrzymanych wymagań	X						
2.4	Ankieta				X			
2.4.1	Przygotowanie ankiety precyzującej wymagania				X			
2.4.2	Przeprowadzenie ankiety				X			
2.4.3	Analiza danych i przygotowanie wniosków z ankiety	X			X			
2.5	Konsultacje ze zleceniodawcą	X						
2.6	Uściślenie wymagań	X						
2.7	Stworzenie dokumentu definicji wymagań	X						
2.8	Przyjęcie ostatecznej wersji wymagań							X
2.9	Grupowanie wymagań użytkownika	X						
2.10	Praca nad modelem przypadków użycia							
2.10.1	Opracowanie ogólnego diagramu use-case	X						
2.10.2	Przedyskutowanie diagramu przypadków użycia	X						
2.10.3	Uszczegółowienie diagramu przypadków użycia	X						X
2.10.4	Przyjęcie modelu przypadków użycia							X
2.11	Praca nad modelem klas	X						X
2.11.1	Identyfikacja klas i obiektów	X						X
2.11.2	Weryfikacja zidentyfikowanych klas	X						
2.11.3	Budowa diagramu klas						X	
2.11.4	Badanie poprawności modelu klas	X					X	X

2.11.5	Przyjęcie modelu klas							X
2.12	Praca nad scenariuszmi trudniejszych funkcji systemu	X			X			X
2.13	Ocena wytworzonego modelu							X
2.14	Koniec fazy analizy							
3	Faza projektowania							
3.1	Identyfikacja funkcji trudnych do zrealizowania							
3.2	Uściślanie danych otrzymanych z fazy analizy							
3.2.1	Identyfikacja funkcji nie związanych z dziedziną projektową							
3.2.2	Uszczegóławianie modelu klas							
3.3	Praca nad GUI							
3.4	Przeniesienie modelu analitycznego na model relacyjny							
3.5	Praca nad optymalizacją projektu							
3.6	Ocena modelu projektowego							
3.7	Przyjęcie modelu projektowego							
4	Faza implementacji							
4.1	Opracowanie dokumentu o standartach programowania							
4.2	Implementacja							
4.3	Testowanie na bieżąco fragmentów kodu							
4.4	Szczegółowe przetestowanie całego systemu							
4.5	Koniec fazy implementacji							
5	Faza instalacji							
5.1	Zainstalowanie systemu u zleceniodawcy							
5.2	Przeprowadzenie szkoleń po stronie klienta							
5.3	Wypełnianie baz danych							
5.4	Przeprowadzenie ostatecznych testów							
5.5	Ostateczne poprawienie błędów							
5.6	Zatwierdzenie systemu przez zleceniodawcę							

3. Struktura organizacji zespołu projektowego

Autor: Tom Szupszyński

Opis: Wydzielenie ról w zespole, określenie ich zakresu i kompetencji

3.1. Wydzielenie ról w zespole projektowym

- Kierownik projektu – Karol Szczepanowski - szczegółowo określa plan działań w projekcie oraz kontroluje postępy prac;
- Analityk – Hubert Sztajerwald – analityk bezpośrednio kontaktuje się z klientem oraz określa wymagania i buduje model systemu;
- Projektant bazy danych/programista – Tom Szupszyński – osoba odpowiedzialna za realizację bazy danych oraz ampie implementację;
- Projektant interfejsu użytkownika/programista – Grzegorz Szolc – osoba odpowiedzialna za realizację interfejsu użytkownika i implementację;
- Osobna wykonująca testy – Andrzej Szuchnicki;
- Osoba odpowiedzialna za konserwację oprogramowania – Michał Bącal;
- Export metodyczny – Michał Ściuba – osoba szczególnie dobrze znająca stosowaną metodykę;

3.2. Struktura organizacji zespołu projektowego

W związku z niewielkim doświadczeniem zespołu projektowego wydaje się być wskazanym zastosowanie gwiazdистой struktury zespołu. Struktura gwiazdista zapewnia kierownikowi dużą kontrolę nad przebiegiem prac projektowym.

4. Standardy komunikacyjne

Autor: Michał Bącal

Opis: Opracowanie standardów komunikacyjnych i wzorca dokumentacji

Zostały przygotowane następujące wzorce dokumentacji:

4.1. Plan Testowania Oprogramowania

1. Wstęp

- 1.1 Wprowadzenie
- 1.2 Zakres
- 1.3 Definicje, Akronimy, Skróty
- 1.4 Referencje, odsyłacze do innych dokumentów
- 1.5 Krótki przegląd

2. Testowane elementy i kryteria

2.1 Testowane elementy

Sekcja identyfikuje testowane elementy i dokumenty, które umożliwią stwierdzenie, w jaki sposób poddawane testom elementy funkcjonują i jakie wykonują operacje.

2.2 Cechy oprogramowania do przetestowania

Część dokumentu, która specyfikuje wszystkie cechy oprogramowania i kombinacje tych cech, które powinny być przetestowane. Lista cech sporządzana jest na podstawie dokumentu wymagań na system lub dokumentu projektu. Cechy powinny być pogrupowane ze względu na numery wersji oprogramowania w przypadku stosowania metody przyrostowej wytwarzania oprogramowania.

2.3 Cechy oprogramowania nie podlegające testom

Lista cech, które nie będą podlegać testom z wyjaśnieniem przyczyn, dla których nie podjęto decyzji o zweryfikowaniu ich.

3. Przyjęte metody

3.1 Podejście

Specyfikacja głównych czynności, metod i narzędzi, które będą użyte do testowania zdefiniowanych wcześniej cech i elementów oprogramowania. Czynności powinny być opisane

bardzo szczegółowo, aby na podstawie tych opisów można było wyróżnić zadania w obrębie procesu weryfikacji oprogramowania, oszacować zasoby i czas.

3.2 Kryteria oceny weryfikowanych cech i elementów

Lista kryteriów, na podstawie których testowane elementy będą kwalifikowane jako elementy, które przeszły lub nie przeszły procesu weryfikacji.

3.3 Kryteria zawieszania i wznowiania testów

Sekcja definiująca kryteria na podstawie, których podejmowana będzie decyzja o zawieszeniu czynności w obrębie testów związanych z weryfikowanymi elementami. Ten fragment dokumentu specyfikuje czynności, które należy wykonać powtórnie w przypadku wznowienia testów.

3.4 Zadania w obrębie testowania

Identyfikacja zadań niezbędnych do wykonania podczas i przed podjęciem decyzji o rozpoczęciu testów. Lista zadań powinna zawierać informacje dotyczące wewnętrznych zależności pomiędzy czynnościami, jak również informacje odnośnie wymagań jakim muszą sprostać osoby będące odpowiedzialne za wykonanie tych zadań.

3.5 Wymagania odnośnie środowiska

Sekcja specyfikuje cechy środowiska testowego, jakie powinny być spełnione przed przystąpieniem do weryfikacji oprogramowania. Na środowisko testowe składa się:

- sprzęt komputerowy, na którym będzie przeprowadzany proces weryfikacji,
- oprogramowanie komunikacyjne (wykorzystywane protokoły komunikacyjne),
- oprogramowanie systemowe (rodzaje systemów operacyjnych),
- narzędzia testowe.

3.6 Podział odpowiedzialności

Określenie osób lub grup osób odpowiedzialnych za zarządzanie, projektowanie, przygotowywanie, wykonywanie i sprawdzanie procesu testowania. Grupy mogą uwzględniać autorów weryfikowanych elementów, przedstawicieli użytkownika, personel operacyjny, administratorów danych, jak również członków personelu zapewnienia jakości.

3.7 Harmonogram

Sekcja powinna zawierać „kamienie milowe” całego procesu testowania wraz z określeniem terminów dostarczania poszczególnych elementów oprogramowania na potrzeby testów integracyjnych i systemowych. Dodatkowymi informacjami, które powinny być umieszczone w tym fragmencie dokumentu, są informacje dotyczące terminów wykonania poszczególnych zadań oraz informacje dotyczące zasobów niezbędnych dla przeprowadzenia weryfikacji.

3.8 Aprobata

W tej części dokumentu powinna być umiejscowiona lista osób, które będą odpowiedzialne za zatwierdzenie niniejszego dokumentu i kontrolę efektów prac.

4.2. Plan Zapewnienia Jakości Oprogramowania

1. Wstęp

- 1.1 Cel
- 1.2 Zakres
- 1.3 Definicje, Akronimy, Skróty
- 1.4 Referencje, odsyłacze do innych dokumentów
- 1.5 Krótki przegląd

2 Standardy, praktyki konwencje i metryki

Opisuje się je detalicznie lub odsyła do innych dokumentów.

- 2.1. Standardy dokumentacyjne
- 2.2. Standardy projektowe
- 2.3. Standardy kodowania
- 2.4. Standardy komentowania
- 2.5. Standardy i praktyki testowania
- 2.6. Wybrane metryki ZJO
- 2.7. Ustalenia dotyczące sposobu monitorowania zgodności z planem

3 Przeglądy i audyty

Identyfikuje techniczne przeglądy, przejścia, inspekcje, audyty mające zastosowanie w tej fazie, oraz cel każdego z nich. Opisuje sposoby monitorowania zgodności tych procedur z planem oraz rolę personelu ZJO w tych procedurach.

4. Testowanie

Opisuje w jaki sposób czynności weryfikacji i walidacji oprogramowania będą monitorowane i w jaki sposób będą sprawdzane testy akceptacyjne

5. Raportowanie problemów i akcje korygujące

Raportowanie problemów i akcje korygujące. Identyfikuje procedury zgłaszania problemów oraz podejmowania akcji mających na celu usunięcie problemów. Może opisywać metryki stosowane do procedur zgłaszania problemów mające wpływ na jakość oprogramowania.

6. Narzędzia, techniki i metody

7. Kontrola kodu

Procedury stosowane do pielęgnacji, przechowywania, zabezpieczania i dokumentowania kodu oprogramowania.

8. Kontrola mediów

Dotyczy procedur stosowanych do do pielęgnacji, przechowywania, zabezpieczania i dokumentowania mediów, na których oprogramowanie i dokumentacja będą przechowywane.

9. Kontrola dostawców

Procedury stosowane do kontroli zewnętrznych organizacji lub osób, które rozwijają lub dostarczają oprogramowanie niezbędne dla projektu. Procedury powinny określać standardy, które mają być stosowane przez dostawców, oraz powinny określać sposoby kontroli przestrzegania tych standardów.

10. Zbieranie, pielęgnacja i utrzymanie zapisów

Identyfikuje procedury stosowane do przechowywania informacji zebranych ze wszelkich aktywności, takich jak spotkania, przeglądy, przejścia, audyty, notatek, korespondencji. Powinny określać gdzie te informacje/dokumenty są przechowywane i jak długo, oraz określać sposób dostępu do tych informacji.

11. Szkolenie

Zawiera informacje o przeprowadzanych szkoleniach, ich charakterze i celu.

12. Zarządzanie ryzykiem

4.3. Dokument Specyfikacji Wymagań

1. Wstęp

- 1.1 Cel
- 1.2 Zakres
- 1.3 Definicje, Akronimy, Skróty
- 1.4 Referencje, odsyłacze do innych dokumentów
- 1.5 Krótki przegląd

2. Ogólny Opis

2.1 Ogólne możliwości projektowanego systemu

Rozdział poświęcony głównym czynnikom, które mają wpływ na produkt i wymagania z nim związane. Treść tego rozdziału nie powinna być poświęcona szczegółowym wymaganiom, ale łatwiejszemu zrozumieniu wymagań odnoszących się do oprogramowania.

2.2 Walory użytkowe i przydatność projektowanego systemu

Podrozdział poświęcony procesom, które powinny być objęte funkcjonalnością systemu. Opis powinien dotyczyć również tej części procesów, które są objęte poprzez funkcjonalność istniejącego oprogramowania.

2.3 Ogólne ograniczenia

Opis istniejących ograniczeń, które mogą stać się przeszkodą w budowie oprogramowania. Informacje, które znajdują się w tym podrozdziale, będą stanowić podstawę do formułowania wymagań związanych z ograniczeniami (ang. constraint requirements)

2.4 Charakterystyka użytkowników

Podrozdział poświęcony użytkownikom, którzy staną się w przyszłości odbiorcami oprogramowania. Charakterystyka powinna objąć takie informacje, jak poziom wykształcenia, znajomość języków obcych, język ojczysty, doświadczenie techniczne. Należy dokonać klasyfikacji użytkowników na podstawie zebranych informacji.

2.5 Środowisko operacyjne

Sekcja poświęcona opisowi środowiska, w którym będzie projektowany system z uwzględnieniem informacji dotyczących bazy programowo-sprzętowej, systemów zewnętrznych, z którymi aktualnie projektowane oprogramowanie powinno współdziałać wymieniając dane.

2.6 Założenie i zależności

Lista założeń dla przyszłego systemu pochodząca od zamawiającego, która w późniejszym czasie przełoży się na konkretne wymagania odnoszące się do oprogramowania.

Niniejszy podrozdział powinien zawierać wnioski pochodzące z przeprowadzonej analizy ryzyka.

3. Specyficzne Wymagania

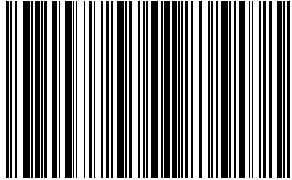
3.1 Wymagania funkcjonalne

3.2 Wymagania нефunkcjonalne

Załączniki

4.4. Uwagi ogólne

Każdy dokument będzie opatrzony następującą metryką:

 Kod dokumentu	Temat: <i>Wpisać temat</i>	Kategoria: <i>Wpisać kategorie dokumentu</i>	Dokument: <i>Podać identyfikator dokumentu</i>
	Autor/Autorzy:: <i>Imiona i nazwiska autorów</i>	Data: <i>Data w formacie dd-mm-rrrr</i>	Wersja: <i>Numer wersji oznaczony kolejną cyfrą 1, 2 itd</i>

Każdy dokument będzie opatrzony unikalnym kodem kreskowym w standardzie Code128 umożliwiające szybsze zarządzanie dokumentami w wersji drukowanej.

5. Ankieta dotycząca badania wymagań

Autor: Hubert Sztajerwald

Opis: Opracowanie i przeprowadzenie ankiety wykonywanej w celu określenia wymagań dla systemu/

5.1. Ankieta

Podczas analizy wymagań pojawiły się następujące nieścisłości:

- W jaki sposób system ma informować skład sędziowski i zawodników?
- Czy juror może mieć mniej niż 18 lat?
- Czy może odbywać się parę zawodów naraz?
- Czy lista zawodników ma jakieś ograniczenie?
- Czy wszyscy zawodnicy muszą być zapamiętani?
- Czy mają być zapamiętane też osoby które już nie są zawodnikami?
- Jak długo mają być zapamiętane wyniki zawodów?
- Czy jest możliwe wystąpienie paru zawodników z jednego kraju?
- Na ile dni przed zawodami zawodnik musi zadeklarować udział?
- Na ile dni przed zawodami musi wpłacić wpisowe?
- Jeżeli zawodnik wpłacił wpisowe i zachoruje, to czy odzyska wpisowe?
- Czy wpisowe obejmuje ubezpieczenie?
- Czy zawodnik może liczyć na pomoc medyczną w razie wypadku?
- Czy zawody mogą odbywać się kilka razy z rzędu na terenie jednego państwa?
- Czy może być więcej niż jeden sponsor?
- Czy ten sam sponsor może sprawować patronat nad zawodami kilka razy z rzędu?
- Czy istnieje ochrona danych osobowych zawodników?
- W ilu kolejnych zawodach juror może wchodzić w skład Jury?
- Czy aktywny zawodnik może wchodzić w skład jury, kiedy nie bierze udziału w żadnych zawodach?
- Czy sędziowie mogą wchodzić w skład zespołu muzycznego?

5.2. Ankieta- odpowiedzi udzielone przez klienta

- W jaki sposób system ma informować skład sędziowski i zawodników?
Odp.: Informować ich za pomocą poczty elektronicznej oraz telefonicznie

- Czy juror może mieć mniej niż 18 lat?

Odp.: Juror musi mieć co najmniej 21lat i odpowiednie kwalifikacje

- Czy może odbywać się parę zawodów naraz?

Odp.: Nie, ponieważ na każdych zawodach startują najlepsi zawodnicy i organizowanie kilku rodzajów zawodów na raz zmniejszyłoby atrakcyjność zawodów i zmniejszyło ich rangę

- Czy lista zawodników ma jakieś ograniczenie?

Odp.: Nie ma ograniczeń

- Czy wszyscy zawodnicy muszą być zapamiętani?

Odp.: Wszyscy którzy brali udział w zawodach i oprócz tego dodatkowo zawodnicy ,którzy zajęli miejsca na podium

- Czy mają być zapamiętane też osoby które już nie są zawodnikami?

Odp.: Tak, ma być przeprowadzona ewidencja wszystkich osób biorących udział w zawodach

- Jak długo mają być zapamiętane wyniki zawodów?

Odp.: Nielimitowany czas...

- Czy jest możliwe wystąpienie paru zawodników z jednego kraju?

Odp.: Tak, pod warunkiem ze spełniają odpowiednie warunki(typu wiek albo określone wyniki, które uprawniają ich do wzięcia udziału)

- Na ile dni przed zawodami zawodnik musi zadeklarować udział?

Odp.: 1 miesiąc

- Na ile dni przed zawodami musi wpłacić wpisowe?

Odp.: 2 tygodnie

- Jeżeli zawodnik wpłacił wpisowe i zachoruje, to czy odzyska wpisowe?

Odp.: Nie

- Czy wpisowe obejmuje ubezpieczenie?

Odp.: Tak

- Czy zawodnik może liczyć na pomoc medyczną w razie wypadku?

Odp.: Zawsze może liczyć na pomoc medyczna

- Czy zawody mogą odbywać się kilka razy z rzędu na terenie jednego państwa?

Odp.: Nie

- Czy może być więcej niż jeden sponsor?

Odp.: Może być maksymalnie 3

- Czy ten sam sponsor może sprawować patronat nad zawodami kilka razy z rzędu?

Odp.: Owszem

- Czy istnieje ochrona danych osobowych zawodników?

Odp.: Niestety nie

- W ilu kolejnych zawodach juror może wchodzić w skład Jury?

Odp.: Jeśli ma odpowiednie kwalifikacji w dowolnej ilości, chce priorytet mają osoby ,które nie sędziowały w poprzednich zawodach

- Czy aktywny zawodnik może wchodzić w skład jury, kiedy nie bierze udziału w żadnych zawodach?

Odp.: Nie żaden zawodnik nie może być w jury

- Czy sędziowie mogą wchodzić w skład zespołu muzycznego?

Odp.: Nie

6. Plan zapewnienia jakości

Autor: Michał Ściuba

Opis: Opracowanie planu działań zmierzającego do zapewnienia jakości produktu końcowego.

6.1. Cel PZJO (plan zarządzania jakością oprogramowania)

Celem PZJO jest ustalenie zasad polityki ZJO tak, aby produkt właściwie realizował swoje funkcje i odpowiadał aktualnym wymaganiom i standardom.

Plan skierowany jest do członków zespołu projektowego.

PZJO odnosi się do następujących faz rozwoju oprogramowania:

- faza wymagań użytkownika i analizy
- faza projektu architektury
- faza projektowania i konstrukcji
- faza budowy, testowania i instalacji oprogramowania

6.2. Zarządzanie

Zadania:

6.2.1 Faza wymagań użytkownika i analizy

1. Sprawdzenie czy wymagania użytkownika są wystarczająco sprecyzowane
2. Sprawdzenie czy plany są zdefiniowane zgodnie ze standardami
3. Certyfikacja systemów przed skierowaniem do produkcji
4. Wymuszanie standardów gromadzenia i przechowywania danych

6.2.2 Faza projektu architektury

1. Sprawdzenie czy procedury są wykonywane zgodnie z planami
2. Sprawdzenie czy zagrożenia projektu są zminimalizowane
3. Sprawdzenie czy członkowie zespołu projektowego mają zdefiniowane zadania i odpowiedzialności

6.2.3 Faza projektowania i konstrukcji

1. Ustalenie czy standardy, praktyki i konwencje są przestrzegane
2. Sprawdzenie czy dokumentacja jest sporządzana odpowiednio
3. Sprawdzenie czy produkty są implementowane zgodnie z planami
4. Sprawdzenie czy projekty używają właściwych narzędzi, technik i metod
5. Sprawdzenie czy oprogramowanie jest przechowywane w kontrolowanych bibliotekach
6. Sprawdzenie czy oprogramowanie jest przechowywane w sposób bezpieczny i chroniony
7. Sprawdzenie czy rejestrowane są wszelkie aktywności związane z oprogramowaniem
8. Sprawdzenie czy oprogramowanie od zewnętrznych dostawców spełnia odpowiednie standardy
9. Sprawdzenie czy problemy są rejestrowane i reakcja na nie jest właściwa

Faza budowy, testowania i instalacji oprogramowania

1. Sprawdzenie czy dane pomiarowe są gromadzone i używane do poprawy produktów i procesów.
2. Sprawdzenie czy testy są specyfikowane i rygorystycznie przestrzegane
3. Ocena i modelowanie wydajności oprogramowania
4. Czuwanie nad tym czy kolejne komponenty spełniają oczekiwania klienta i czy ewentualne uwagi są przestrzegane przy implementacji kolejnej części

6.3. Przeglądy i audyty

Jako że celem audytu jest dostarczenie odbiorcy i dostawcy obiektywnych, aktualnych i syntetycznych informacji o stanie całego projektu powinien on być wykonywany przez niezależną firmę audytorską.

Wyniki audytu przedstawiane są najpierw kierownikowi a następnie zleceniodawcy.

6.4. Kontrola kodu

6.5. Kontrola mediów

Kod będzie przechowywany na jednym wspólnym serwerze, który stanowi scentralizowaną bazę kodów źródłowych. Za bezpieczeństwo, archiwizację oraz utrzymanie odpowiedzialny jest pracownik firmy – administrator sieci.

6.6. Kontrola dostawców

6.7. Zbieranie, pielęgnacja i utrzymywanie zapisów

Wszelkie aktywności jak spotkania, przeglądy, przejścia, audyty, notatki, korespondencja, wykryte błędy i ich korekty powinny być dokumentowane.

Raz w miesiącu zebrane dokumenty są wypalane na płytach CD, każda płyta nagrywana jest podwójnie, aby zapobiec ewentualnej utracie danych.

Płyty archiwizowane są w specjalnie przeznaczonym pomieszczeniu, do którego dostęp ma tylko jeden pracownik firmy, odpowiedzialny za utrzymanie archiwum.

Dokumentacja przechowywana jest przez pięć lat od momentu archiwizacji.

7. Plan testowania oprogramowania

Autor: Andrzej Szuchnicki

Opis: Opracowanie planu testowania oprogramowania.

Głównym celem testowania jest wykrycie jak największej liczby błędów w programie i ocena niezawodności oprogramowania. Oprogramowanie będzie testowane w sposób dynamiczny, który polega na wykonywaniu fragmentów programu i porównywaniu uzyskanych wyników z wynikami poprawnymi. W naszych testach oprzemy się na weryfikacji systemu, czyli na testowaniu zgodności systemu z wymaganiami zdefiniowanymi w fazie określenia wymagań.

Weryfikacja włącza następujące czynności:

1. Przeglądy techniczne oraz inspekcje oprogramowania.
 - 1.1. Przegląd techniczny (projektu roboczy lub zbiór projektów roboczych). Wszyscy zainteresowani projektem oceniają elementy oprogramowania na zgodność postępu prac z przyjętym planem.
 - 1.2. Inspekcje oprogramowania. (kiedy?, po zgłoszeniu potrzeby). Wymagania na oprogramowanie, projekt lub kod są szczegółowo badane przez osobę lub grupę osób nie będących autorami, w celu identyfikacji błędów, naruszenia standardów i innych problemów.
2. Sprawdzenie, czy wymagania na oprogramowanie są zgodnie z wymaganiami użytkownika.
 - 2.1. Sprawdzenie czy interfejsy systemu są zgodne z wymaganiami określonymi przez użytkowników.
3. Sprawdzenie czy komponenty projektu są zgodne z wymaganiami na oprogramowanie. Właściwości operacyjne systemu, np. wymagania logistyczne, organizacyjne, użyteczność, czytelność ekranów, operacje wymagające zbyt wielu kroków, jakość komunikacji systemu, jakość informacji o błędach, jakość pomocy.
4. Testowanie jednostek oprogramowania(modułów).(wykonanie tej czynności po zakończeniu każdego z etapów implementacji).
 - 4.1. Test strukturalny.
Wstawienie kodu diagnostycznego i użycie debuggerów w celu analizy krok po kroku.
Przygotowanie danych testowych lub użycie specjalnych programów usprawniających testowanie.
 - 4.2. Test funkcjonalny. Test powinny wykonać osoby trzecie.
Testowanie funkcji systemu.
5. Testowanie integracji oprogramowania, testowanie systemu. W tej fazie integrowane są poszczególne moduły i testowane są podsystemy oraz systemy jako całość.
 - 5.1. Testy zużycia zasobów: zużycie czasu jednostki centralnej, zużycie pamięci operacyjnej, przestrzeni dyskowej.
 - 5.2. Zabezpieczenia systemu :
 - 5.2.1.1. zabezpieczenia haseł użytkowników
 - 5.2.1.2. testy zamykania zasobów przed niepowołanym dostępem
 - 5.2.1.3. testy dostępu do plików przez niepowołanych użytkowników
 - 5.3. Przenoszalność oprogramowania: czy oprogramowanie będzie działać w zróżnicowanym środowisku, rozdzielczościach ekranów.

- 5.4. Niezawodność oprogramowania, mierzona średnim czasem pomiędzy błędami. Miary niezawodności:
 - 5.4.1. Prawdopodobieństwo błędnego wykonania podczas realizacji transakcji. Każde błędne wykonanie powoduje zerwanie całej transakcji. Miarą jest częstość występowania transakcji, które nie powiodły się wskutek błędów.
 - 5.4.2. Częstość wystąpienia błędnych wykonań: liczba błędów w jednostce czasu.
- 5.5. Odtwarzalność oprogramowania, mierzona średnim czasem reperowania oprogramowania po jego awarii.
- 5.6. Kompletność i jakość złożonych funkcji systemu.
- 5.7. Nie przekraczanie ograniczeń, np. zajmowaną pamięć.
- 5.8. Obciążenie oprogramowania, tj. jego zdolność do poprawnej pracy przy ekstremalnie dużych obciążeniach. Np. maksymalnej liczbie użytkowników, bardzo dużych rozmiarach plików, maksymalnych zapisach, bardzo długich liniach danych źródłowych
- 5.9. Skalowalność systemu, tj. spełnienie warunków przy znacznym wzroście obciążenia.

8. Oszacowanie złożoności projektu

Autorzy: Karol Szczepanowski i Michał Bącał

Opis: Analiza złożoności projektu metoda FPA

8.1. Analiza Złożoności Oprogramowania Metodą Zmiennych Funkcyjnych (FPA)

8.1.1 Nie skorygowane punkty funkcyjne

Liczbę nie skorygowanych punktów funkcyjnych wylicza się na podstawie tabeli korzystając z poniższych danych:

Wejścia Użytkownika- *obiekty wejściowe wpływające na dane w systemie.*

Wyjścia Użytkownika – *obiekty związane z danymi w systemie.*

Zbiory Danych Wewnętrzne – *liczba wewnętrznych plików roboczych.*

Zbiory Danych Zewnętrzne – *liczba plików zewnętrznych zapełnianych przez produkt programowy.*

Zapytania Zewnętrzne – *interfejsy z otoczeniem programu*

Liczba nie skorygowanych punktów FP:

5 3

$$UFP = \sum_{I=1} \sum_{j=1} w_{ij} * n_{ij}$$

Gdzie w_{ij} to waga a n_{ij} liczba elementów

Wagi przypisywane projektom			
	Projekt prosty	Projekt średni	Projekt złożony

I=1	Wejścia użytkownika	3	4	6
I=2	Wyjścia użytkownika	4	5	7
I=3	Zbiory danych wewnętrzne	7	10	15
I=4	Zbiory danych zewnętrzne	5	7	10
I=5	Zapytania zewnętrzne	3	4	6

8.1.2 Korekcja Punktów Funkcyjnych

- Występowanie urządzeń komunikacyjnych
- Rozproszenie przetwarzania
- Długość czasu oczekiwania na odpowiedź systemu
- Stopień obciążenia sprzętu istniejącego
- Częstotliwość wykonywania dużych transakcji
- Wprowadzanie danych w trybie bezpośrednim
- Wydajność użytkownika końcowego
- Aktualizacja danych w trybie bezpośrednim
- Złożoność przetwarzania danych
- Możliwość ponownego użycia programów w innych zastosowaniach
- Łatwość instalacji
- Łatwość obsługi systemu
- Rozproszenie terytorialne
- Łatwość wprowadzania zmian - pielęgnowania systemu

Wpływ powyższego czynnika:

Bez wpływu = 0

Bardzo silny wpływ = 5

14

Kompleksowy współczynnik korygujący $VAF = \sum K_k$

$K=1$

3. Punkty Funkcyjne (FPs):

$FP = (0.65 + 0.01 * VAF) * UFP$

$FP = (0.65...1.35) * UFP$

4. Plan Szacowania Złożoności Metodyką FPA

- Identyfikacja systemu
- Obliczenie współczynnika korygującego
- Wyznaczenie ilości zbiorów danych i ich złożoności
- Wyznaczenie ilości i złożoności elementów funkcjonalnych (we, wy, zapytania)
- Realizacja obliczeń
- Weryfikacja
- Raport, zebranie recenzujące

8.2. Obliczenie punktów funkcyjnych

Elementy	Proste	Waga	Średnie	Waga	Złożone	Waga	Razem
Wejścia	3	3	6	4	2	6	45
Wyjścia	7	4	6	5	6	7	100
Zbiory wew.	4	7	4	10	1	15	83
Zbiory zew.	0	5	2	7	0	10	14
Zapytania	12	3	7	4	0	6	78
						Łącznie:	320

Szacunkowa złożoność projektu wynosi ok. 320 FP