

# Overlord - Software Development Plan i Plan Testów

# Cel prezentacji

- przedstawienie kwestii związanych z pracami przy projekcie Overlord
- przedstawienie planu testowania systemu

# Organizacja pracy

Rola	Osoba	Zadania
Kierownik Zespołu	Tomasz Weksej	<ul style="list-style-type: none"><li>• kontrola i podział pracy zespołu</li><li>• kontakty zewnętrzne</li></ul>
Projektant, specjalista ds. eksploatacji	Jakub Gołębiowski	<ul style="list-style-type: none"><li>• definiuje cykl życia planu, oraz wszelkie koncepcje eksploatacyjne</li></ul>
Projektant, specjalista ds. wymagań	Adam Kawa	<ul style="list-style-type: none"><li>• definiuje cykl życia planu, oraz wymagania systemowe</li></ul>
Projektant, specjalista ds. architektury	Piotr Krewski	<ul style="list-style-type: none"><li>• definiuje wszelkie koncepcje dotyczące architektury oprogramowania i systemu</li></ul>
Programista	Jakub Gołębiowski, Adam Kawa, Piotr Krewski, Tomasz Weksej	<ul style="list-style-type: none"><li>• odpowiada za implementację i działanie przydzielonej mu części aplikacji</li></ul>
Tester	Jakub Gołębiowski, Adam Kawa, Piotr Krewski, Tomasz Weksej	<ul style="list-style-type: none"><li>• planuje, przeprowadza i raportuje testy</li><li>• wykrywa błędy krytyczne i niezgodności ze specyfikacją</li></ul>

# Kamienie milowe

- Początek
- Opracowanie zasad gry
- Zatwierdzenie dokumentacji (Wizja, Przypadki użycia)
- Zatwierdzenie dokumentacji (SDP, SAD)
- Zaimplementowanie modelu
- Wydanie ostatecznej wersji

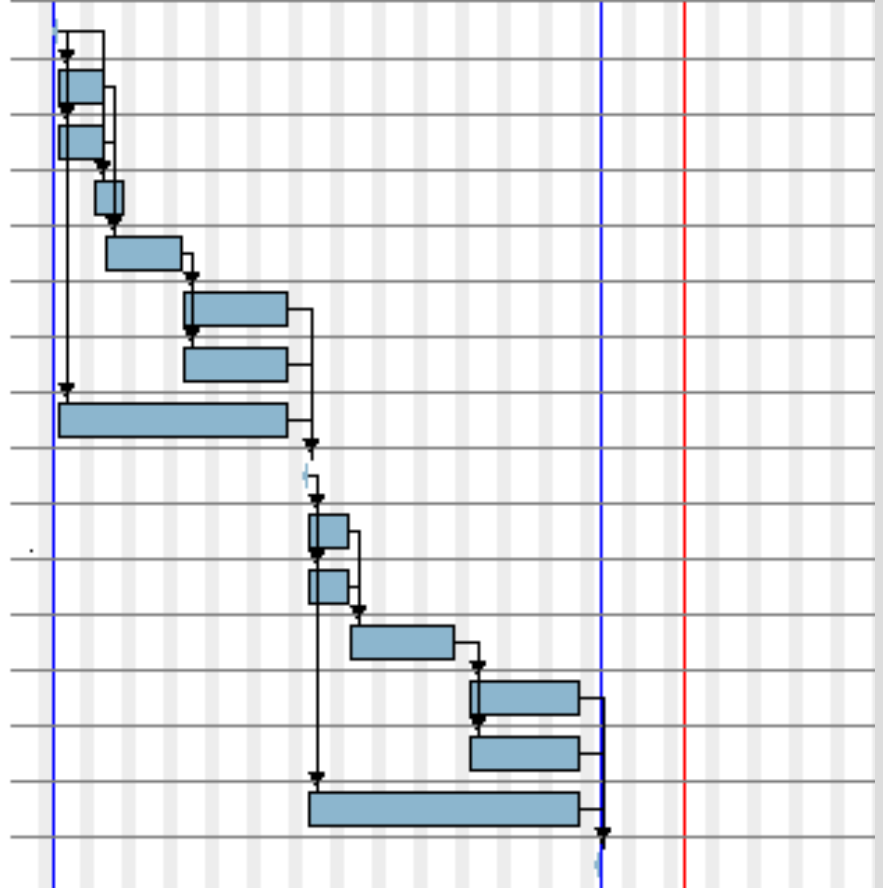
# Faza projektowa



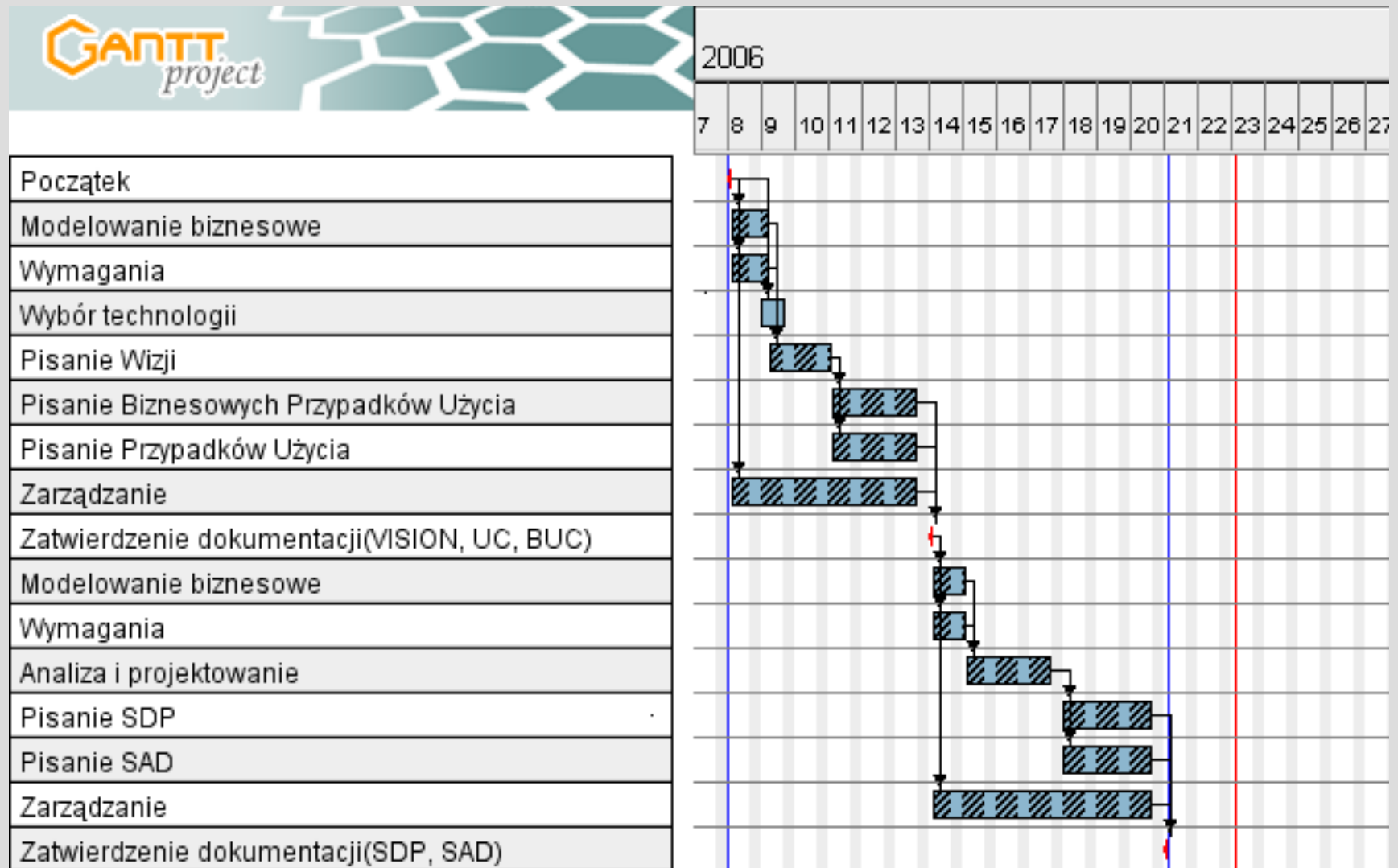
2006

7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Początek
Modelowanie biznesowe
Wymagania
Wybór technologii
Pisanie Wizji
Pisanie Biznesowych Przypadków Użycia
Pisanie Przypadków Użycia
Zarządzanie
Zatwierdzenie dokumentacji(VISION, UC, BUC)
Modelowanie biznesowe
Wymagania
Analiza i projektowanie
Pisanie SDP
Pisanie SAD
Zarządzanie
Zatwierdzenie dokumentacji(SDP, SAD)

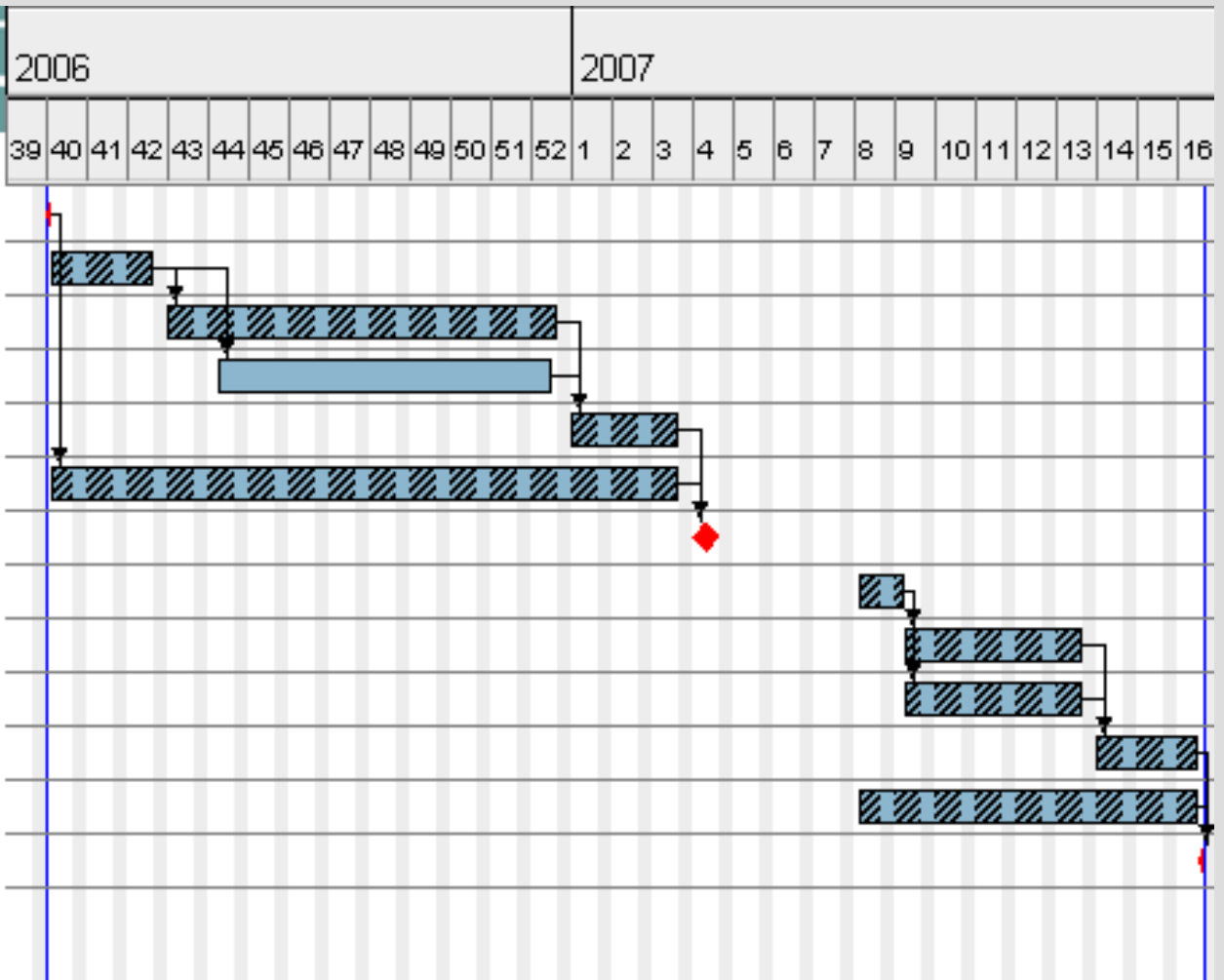


# Faza projektowa - ścieżka krytyczna





# Faza implementacyjna - ścieżka krytyczna



# Nadzór i kontrola projektu

- Plan zarządzania wymaganiami
- Plan zarządzania harmonogramem
- Plan kontroli jakości
- Plan zarządzania ryzykiem

# Plan kontroli jakości

Rodzaj	Częstotliwość	Cel
przeгляд spójności dokumentacji	raz na 3 tygodnie	wykrycie sprzeczności w dokumentacji
przeгляdy postępu	raz na 12 tygodni	potwierdzenie przebiegu prac zgodnie z harmonogramem, wyjaśnianie wątpliwości, wzajemna mobilizacja do pracy
testowanie	równoległe z implementacją, również po ukończeniu pierwszej wersji systemu	wykrycie błędów
walidacja	raz na 23 tygodnie	wykrycie rozbieżności między implementacją a założeniami i dokumentacją

# Plan kontroli ryzyka

Nazwa	Skutki	Stopień ryzyka	Przeciwdziałanie
nadmiar obowiązków związanych z uczelnią	opóźnienia w pracy	średni	mądra organizacja czasu, przeglądy postępu
problem z opanowaniem technologii	trudności z realizacją powierzonych zadań	niski	konsultacje w obrębie zespołu oraz zewnętrzne, wymiana zadań, stały kontakt
konflikty pomiędzy członkami zespołu	paraliż prac	niski	regularna integracja członków zespołu
słaby przepływ informacji	opóźnienia	niski	spotkania na zajęciach, używanie telefonów komórkowych, emaili, komunikatorów internetowych

# Cele systemu testów

- zapewnienie jakości wytwarzanego software'u
- sprawdzenie zgodności ze specyfikacją
- zweryfikowanie funkcjonalności poszczególnych części systemu i ich wzajemnej kooperacji

# Zakres testów

- Testy funkcjonalności
- Testy interfejsu użytkownika
- Testy integralności bazy danych
- Testy wydajności i obciążenia
- Testy bezpieczeństwa i kontroli dostępu
- Testy konfiguracji
- Testy regresyjne

# Testy funkcjonalności

Cel testu	Dowiedzenie właściwej funkcjonalności systemu, włączając w to nawigację, wprowadzanie, przetwarzanie, odzyskiwanie, aktualizację i wyszukiwanie danych.
Technika	Zastosowanie każdego przypadku użycia, używając zarówno prawidłowych, jak i nieprawidłowych danych, by zweryfikować: <ul style="list-style-type: none"><li>• czy wprowadzenie prawidłowych danych spowoduje pojawienie się spodziewanych rezultatów</li><li>• czy wprowadzenie nieprawidłowych danych spowoduje wyświetlenie się odpowiednich komunikatów i nie spowoduje niestabilności systemu</li><li>• czy serwis zaspokaja wszystkie przypadki użycia</li></ul>
Testowane komponenty	Model, Interfejs Użytkownika

# Testy interfejsu użytkownika

Cel testu	Celem testu Interfejsu użytkownika jest zweryfikowanie czy UI udostępnia użytkownikowi właściwy dostęp do funkcji systemu.
Technika	Stworzenie testów dla każdego okna
Testowane komponenty	Interfejs Użytkownika

# Testy integralności bazy danych

Cel testu	Zapewnienie, że funkcje i metody dostępu nie naruszają integralności BD, ani nie powodują zniszczenia danych
Technika	<ul style="list-style-type: none"><li>• Odwoływanie się do Bazy Danych z poprawnymi i niepoprawnymi zestawami danych, lub zapytaniami o dane</li></ul>
Testowane komponenty	Baza danych

# Testy wydajności i obciążenia

Cel testu	Pokazanie, że system jest w stanie obsłużyć wiele zapytań w zadowalającym czasie i zapewniając właściwą izolację transakcji
Technika	<ul style="list-style-type: none"><li>• Zasymulowanie jednoczesnego korzystania z systemu przez 100 000 użytkowników.</li></ul>
Testowane komponenty	Model, Baza Danych

# Testy bezpieczeństwa i kontroli dostępu

Cel testu	<ul style="list-style-type: none"><li>• Bezpieczeństwo aplikacji: sprawdzenie, czy użytkownik ma dostęp jedynie do tych funkcji, które są dla niego przeznaczone.</li><li>• Bezpieczeństwo systemu: weryfikacja, czy tylko użytkownicy posiadający bezpośredni dostęp do systemu oraz aplikacje mają dostęp do niego.</li></ul>
Technika	<ul style="list-style-type: none"><li>• Wylistowanie, dla każdego typu użytkownika, funkcji które ma on prawo wykonać</li><li>• Stworzenie testów dla każdego typu użytkownika i sprawdzenie każdego prawa dostępu</li></ul>
Testowane komponenty	Model, Interfejs Użytkownika, Baza Danych

# Testy konfiguracji

Cel testu	Testy te mają na celu zapewnienie poprawnego zachowania się systemu na rozmaitych konfiguracjach sprzętowych - zwłaszcza różnych przeglądarkach internetowych.
Technika	Wykoanie skryptów testowych na różnych przeglądarkach internetowych i ich konfiguracjach
Testowane komponenty	Interfejs użytkownika

# Testy regresyjne

Cel testu	Sprawdzenie czy poszczególne części systemu właściwie ze sobą współpracują po dokonanych zmianach
Technika	Przeprowadzenie tych samych testów, które były dokonane przed każdą iteracją i porównanie wyników
Testowane komponenty	Wszystkie

# Harmonogram testów

- włączenie do testów bazy danych 01.11.2006
- testy integralności bazy danych 01.11.2006 - 10.11.2006
- włączenie do testów modelu 10.11.2006
- testy wydajności i obciążenia 11.11.2006 - 05.12.2006
- testy regresyjne 05.12.2006 - 15.12.2006
- włączenie do testów interfejsu użytkownika 16.12.2006
- testowanie interfejsu użytkownika 16.12.2006 - 01.01.2007
- testy konfiguracji 02.01.2007 - 15.01.2007
- testy regresyjne 15.01.2007 - 01.02.2007
- testy funkcjonalności 02.02.2007 - 17.02.2007
- testy bezpieczeństwa i kontroli dostępu 17.02.2007 - 01.03.2007
- finalne testy regresyjne 01.03.2007 - 01.04.2007

# Zasoby ludzkie

- Kierownik testów
- Projektant testów
- Tester

# Zasoby programowe

- Systemy operacyjne:
  - Windows
  - Linux
- Przeglądarki WWW:
  - Firefox
  - Opera
  - Mozilla
  - Netscape

**Dziękujemy za uwagę!**