

Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych
Instytut Informatyki

Rok akademicki 2009/2010

Praca dyplomowa inżynierska

Piotr Kalański

System Service Desk zgodny z zaleceniami ITIL

Opiekun pracy:
dr inż. Michał Rudowski

Ocena:

**Podpis Przewodniczącego
Komisji Egzaminu Dyplomowego**



Kierunek:	Informatyka
Specjalność:	Inżynieria Systemów Informatycznych
Data urodzenia:	18 kwietnia 1986 r.
Data rozpoczęcia studiów:	październik 2006 r.

Życiorys

Nazywam się Piotr Kalański. Urodziłem się 18 kwietnia 1986 r. w Warszawie. W mieście rodzinnym ukończył szkołę podstawową nr 2 im. Janusza Korczaka oraz gimnazjum nr 1 im. Hugona Kołłątaja. W 2002 roku rozpoczęłem naukę w XL LO im. Stefana Żeromskiego w Warszawie w klasie o profilu matematyczno-informatycznym.

W 2005 roku rozpoczęłem studia w Szkole Głównej Gospodarstwa Wiejskiego w Warszawie na Wydziale Zastosowań Matematyki i Informatyki, kierunek Informatyka i Ekonometria. W tym okresie informatyka stała się kluczowym obszarem moich zainteresowań. W 2006 roku rozpoczęłem studia na Politechnice Warszawskiej na Wydziale Elektroniki i Technik Informacyjnych, kierunek Informatyka.

Od lipca 2008 roku do października 2008 roku uczestniczyłem w Programie Praktyk Letnich w Banku Handlowym S.A. Zadaniem wykonywanym w czasie odbywania praktyk było rozwijanie Portalu Operacji Kredytowych i Leasingowych. Od lutego 2009 roku jestem pracownikiem Kredyt Banku S.A. W organizacji tej pracowałem na następujących stanowiskach: Analityk MIS, Menedżer Aplikacji. Jestem odpowiedzialny za System Wsparcia Sprzedaży.

Moje obszary zainteresowań to inżynieria oprogramowania, bazy danych, CRM, Business Intelligence.

Podpis studenta

Egzamin dyplomowy

Złożył egzamin dyplomowy w dn. 2010 r.

z wynikiem

Ogólny wynik studiów

Dodatkowe wnioski i uwagi Komisji

.....
.....

Streszczenie

W niniejszej pracy przedstawiono projekt i implementację systemu wsparcia realizacji usług informatycznych w zakresie funkcji Service Desk, zgodnego z zaleceniami ITIL oraz złożonego z bazy danych i aplikacji. Została opisana biblioteka ITIL, która jest dziś najpopularniejszym standardem zarządzania usługami IT, ze szczególnym uwzględnieniem wybranych procesów: zarządzania incydentami oraz zarządzania problemami, które zaimplementowano w zrealizowanym systemie. Przedstawiono implementację systemu, obejmującą omówienie wybranych technologii, wykorzystanych narzędzi oraz pakietów klas. Na potrzeby pracy zostały wykrojone dane przykładowej firmy dostarczającej usługi IT, zrealizowane zgodnie z zaleceniami ITIL. Na tym przykładzie przedstawiono wsparcie systemu dla działalności tej firmy. Opisano również przykładowe testy jednostkowe, funkcjonalne oraz wydajnościowe systemu, które zostały utworzone przez autora w ramach realizacji pracy.

Słowa kluczowe: ITIL, ITSM, Service Desk, Incident Management, Problem Management.

Abstract

Title of paper: *Service Desk consistent with ITIL recommendations*

This thesis presents design and implementation of system that supports realization of IT services in Service Desk function, which is consistent with ITIL recommendations and composed of database and application. ITIL library, which is the most popular standard for IT Service Management, has been described with special treatment of chosen processes: incident management and problem management, which have been implemented in created system. Implementation of system, which includes presentation of chosen technologies, used tools and class packages, has been realized and presented. Data of example company, which delivers IT services, has been created. This example shows how implemented system supports this company. Finally, example unit tests, functional tests and performance tests of system has been introduced, which have been created by author.

Keywords: ITIL, ITSM, Service Desk, Incident Management, Problem Management.

Spis treści

Spis treści	i
1 Wstęp	1
1.1 Cel pracy	1
1.2 Zakres pracy	2
2 ITIL	3
2.1 Oszczędności w IT	4
2.1.1 Podział kosztów	4
2.1.2 ITSM a oszczędności	4
2.2 Modele ITSM i ich zastosowania	4
2.3 Korzenie ITIL [6]	6
2.4 Help Desk a Service Desk [16]	6
2.5 ITIL 2 - procesy	7
2.5.1 Service Support	7
2.5.2 Service Delivery	8
2.6 ITIL 3 - obszary	9
2.6.1 Strategia usług – Service Strategy [17]	10
2.6.2 Projektowanie usług – Service Design [15]	10
2.6.3 Przekazanie usług – Service Transition [18]	11
2.6.4 Eksplotacja usług – Service Operation [16]	12
2.6.5 Ustawiczne doskonalenie usług – Continual Service Improvement [14]	13
2.7 Szczegółowe omówienie wybranych procesów	14
2.7.1 Zarządzanie incydentami [16]	14
2.7.2 Zarządzanie problemami [16]	18
2.7.3 Zarządzanie zdarzeniami [16]	21
2.7.4 Zarządzanie zleceniami [16]	23
3 Firma	24

3.1	Opis ogólny	24
3.2	Misja	24
3.3	Cele	24
3.4	Profil firmy	25
3.5	Usługi	25
3.6	Organizacja firmy	29
3.7	Kooperanci	31
3.8	Klienci	32
3.9	Obsługa klienta	33
4	Wymagania	35
4.1	Wymagania funkcjonalne	35
4.1.1	Zarządzanie incydentami	35
4.1.2	Zarządzanie problemami	36
4.1.3	Zarządzanie zdarzeniami	36
4.1.4	Zarządzanie zleceniami	36
4.2	Wymagania niefunkcjonalne	36
4.2.1	NF.WOL – Wolumetria	36
4.2.2	NF.SPR – Sprawność i efektywność	37
4.2.3	NF.SEC – Bezpieczeństwo (security)	37
4.2.4	NF.SAF – Bezpieczeństwo (safety)	37
4.2.5	NF.NZW – Niezawodność i dostępność	37
4.2.6	NF.ERG – Ergonomia	38
4.2.7	NF.STA – Standardy używane przez system	38
4.2.8	NF.PRZ – Przenośność	38
4.2.9	Konkluzja	38
4.3	Elementy realizujące wymagania	38
4.3.1	Zarządzanie zdarzeniami	39
4.3.2	Zarządzanie incydentami	39
4.3.3	Zarządzanie problemami	40
4.3.4	Zarządzanie zleceniami	41
4.3.5	Administracja	41
4.3.6	Zarządzanie pracownikami	42
4.3.7	Zarządzanie sygnałami	42
5	Projekt	44
5.1	Baza danych	44
5.1.1	Zarządzanie incydentami	44
5.1.2	Zarządzanie problemami	46
5.1.3	Zarządzanie zdarzeniami	48
5.1.4	Zarządzanie zleceniami	50
5.1.5	Zarządzanie sygnałami	51

5.1.6	Zarządzanie pracownikami	53
5.1.7	Zarządzanie katalogiem usług	54
5.1.8	Zarządzanie poziomem usług	55
5.2	Aplikacja	56
5.2.1	Warstwy aplikacji	56
5.2.2	Przykładowy diagram sekwencji	56
5.2.3	Aspekty aplikacji	58
5.3	Sygnały	59
5.3.1	Typy sygnałów	60
5.3.2	Rodzaj odbiorcy	60
5.3.3	Generacja sygnałów przy zmianie stanu	61
5.3.4	Generacja sygnałów po upływie czasu	61
5.3.5	Konstrukcja zapytania dla generatora	61
5.3.6	Wysyłanie wiadomości e-mail	63
6	Implementacja	64
6.1	Technologie	64
6.2	Wykorzystane narzędzia	66
6.3	Pakiety aplikacji	67
6.3.1	sd.im	67
6.3.2	sd.pm	68
6.3.3	sd.em	70
6.3.4	sd.rf	72
6.3.5	sd.signal	73
7	Wsparcie systemu dla firmy Red Host S.A.	76
7.1	Scenariusz użycia na poziomie biznesowym	76
7.2	Scenariusz użycia na poziomie systemowym	77
7.2.1	Zgłoszenie incydentu	77
7.2.2	Przypisanie incydentu	78
7.2.3	Karta incydentu	79
7.2.4	Eskalacja	79
7.2.5	Powiadomienie pracownika drugiej linii	80
7.2.6	Historia incydentu	80
7.2.7	Dodanie problemu	81
7.2.8	Karta problemu	81
7.2.9	Analiza problemu	82
7.2.10	Rozwiążanie tymczasowe	82
7.2.11	Rozwiążanie incydentu	83
7.2.12	Zamknięcie incydentu	83
7.3	Podsumowanie	84

8 Testy i ocena	86
8.1 Środowiska testowe	86
8.2 Testy jednostkowe	87
8.3 Przykładowe testy funkcjonalne	88
8.3.1 Cykl życia incydentu	88
8.3.2 Ocena testów funkcjonalnych	90
8.4 Testy wydajnościowe	90
8.4.1 Metoda mierzenia czasu	90
8.4.2 Testy wydajnościowe	91
9 Podsumowanie	93
Bibliografia	96
A Zawartość płyty CD	98
B Instalacja	99
B.1 Pliki konfiguracyjne	99
B.1.1 jdbc.properties	99
B.1.2 messages.properties	99
B.2 Skrypty SQL	99
B.2.1 ddl.sql	99
B.2.2 inserts.sql	99
B.3 Zainstalowanie aplikacji	100
B.3.1 Zainstalowanie aplikacji na serwerze Tomcat przy pomocy Tomcat Manager	100
Spis symboli i skrótów	101
Spis rysunków	102
Spis tabel	104

Rozdział 1

Wstęp

W dzisiejszym czasach informatyka stała się głównym budowniczym nowoczesnego modelu działania w biznesie oraz jednym z najważniejszych elementów innowacyjności. Informatyka wrosła w mechanizmy funkcjonowania organizacji. Ze względu na poziom skomplikowania, powstało wiele nowych problemów. Przede wszystkim powstało pytanie, czy wdrażanie systemów IT, a w konsekwencji ich utrzymywanie jest jedyną perspektywą postrzegania zadań IT. Istnieje potrzeba posiadania modeli pozwalających racjonalizować inwestycje. Potrzebne są wzorce oraz praktyki pozwalające dopasować IT oraz Biznes.

Dostarczanie usług IT to podejście adresujące przytaczane wyżej wyzwania. Dziedzina ta została opisana w ramach zbioru dobrych praktyk IT (ITIL - Information Technology Infrastructure Library). Omówienie tej biblioteki znajduje się w rozdziale 2.

System Service Desk jest najważniejszym elementem w bibliotece ITIL. Jest to punkt kontaktu dla klientów oraz interfejs dla zdefiniowanych procesów. W uproszczeniu, jest to rozbudowany Help Desk, który dodatkowo jest interfejsem dla procesów ITIL związanych z dostarczaniem i wsparciem usług informatycznych.

Na rynku dostępnych jest wiele gotowych rozwiązań Service Desk. Są to głównie systemy komercyjne, cechujące się wysoką ceną licencji. Zostały stworzone z myślą o dużych firmach i korporacjach. Niestety na rynku trudno znaleźć skuteczne rozwiązania w tym zakresie adresowane do mniejszych firm, które nie mogą pozwolić sobie na ponoszenie tak wysokich kosztów zakupu licencji, wdrożenia i eksploatacji.

1.1 Cel pracy

Celem pracy jest zaprojektowanie oraz zaimplementowanie prototypu systemu Service Desk w oparciu o ITIL v3. W ramach pracy autor opisał przykładową fir-

mę dostarczającą usługi IT. Na tym przykładzie przedstawiono wsparcie systemu dla działalności tej firmy.

1.2 Zakres pracy

Zakres pracy jest określony poprzez procesy ITIL. Autor skupił się na następujących procesach: zarządzanie incydentami, zarządzanie problemami, zarządzanie zdarzeniami i zarządzanie zleceniami. W pracy zostały pominięte dwa bardzo ważne procesy: zarządzania konfiguracją i zarządzania zmianami. Procesy te są przedmiotem równolegle realizowanej pracy dyplomowej inżynierskiej Adriana Wiśniewskiego. Fragment systemu zrealizowany w niniejszej pracy będzie współpracować¹ z fragmentem systemu zrealizowanym przez Adriana Wiśniewskiego. Prace analityczne, wykreowanie danych przykładowej firmy oraz konfiguracja aplikacji była realizowana wspólnie. Obie części systemu łącznie będą stanowić system dostarczania i wspierania usług IT zgodny z zaleceniami ITIL.

¹Autor niniejszej pracy użył czasu przyszłego, ponieważ fragment systemu autorstwa Adriana Wiśniewskiego zrealizowany jest z opóźnieniem.

Rozdział 2

ITIL

W wielu organizacjach istnieje problem relacji między Biznesem a IT. Z jednej strony Biznes zarzuca IT, że działania podejmowane przez nich są zbyt kosztowne oraz wymagają długiego czasu realizacji. Z drugiej strony informatycy twierdzą, że Biznes nie potrafi jasno określić swoich potrzeb, a w sytuacji, gdy uda się dojść do porozumienia, koncepcja nagle ulega zmianie. Biznes oczekuje rezultatów w czasie nie możliwym do realizacji. Są to problemy powszechnie występujące, dlatego potrzebne jest podejście umożliwiające poprawienie sytuacji. W tym celu powstały standardy ITSM¹ [20].

W podejściu klasycznym dział IT postrzegany jest jako centrum kosztów. IT skupia się na dostarczaniu systemów, komponentów oraz technologii. Dominuje podejście personalne. W celu wykonania określonego działania należy zwrócić się do konkretnej osoby. Często zdarza się, że bez żadnego uzasadnienia biznesowego wdraża się nowe technologie tylko dlatego, że są nowością na rynku. Stare systemy utrzymywane są tylko dlatego, że działają.

Natomiast w podejściu ITSM nie dostarcza się systemów tylko usługi, w taki sam sposób jak robią to jednostki biznesowe. IT nie jest centrum kosztów, tylko wsparciem dla generowania przychodów. Nie można zwracać się personalnie do konkretnej osoby, należy żądania kierować do grup. Dzięki temu unika się przejęcia najlepszych pracowników oraz umożliwia rozwój nowym pracownikom. Technologie zmieniane są tylko wtedy, gdy ma to uzasadnienie biznesowe i jest poprzedzone rzetelną analizą, w celu określenia kosztów oraz zysków [3].

¹ITSM - Information Technology Service Management

2.1 Oszczędności w IT

2.1.1 Podział kosztów

Koszty bezpośrednie:

CAPEX – koszty związane z inwestycją. Są to koszty projektu, wdrożenia, zakupu nowego sprzętu oraz koszty integracji.

OPEX – koszty związane z eksploatacją. Koszty HR², naprawy sprzętu, szkoleń i licencji.

Koszty pośrednie:

- koszty komunikacji;
- koszty przestojów - straty;
- kary;

2.1.2 ITSM a oszczędności

ITSM umożliwia oszczędzanie w wielu obszarach.

Ograniczanie czasu niedostępności usług IT poprzez szybką naprawę oraz zapobieganie.

Kontrola zmian w infrastrukturze. Każda zmiana jest szczegółowo analizowana przez komitet specjalistów, który sprawdza zasadność zmiany oraz dokonuje oszczędzania kosztów.

W ramach procesu zarządzania pojemnością dąży się do osiągnięcia równowagi pomiędzy zapotrzebowaniem a tym co można zaoferować. Z jednej strony zapobiega się sytuacji, w której klienci nie są w stanie korzystać z usługi, a z drugiej nie można dopuścić do sytuacji, w której nie wykorzystuje się dostępnych zasobów.

Projektowanie usług pod kątem użyteczności biznesowej.

2.2 Modele ITSM i ich zastosowania

Powstało wiele modeli ITSM. Każdy z nich może być stosowany w różnych obszarach.

²HR - Human Resources

COBIT – Control Objectives for Information and related Technology

- zarządzanie taktyczne;
- zarządzanie strategiczne;

CMMI – Capability Maturity Model Integration

- projekty;
- zarządzanie taktyczne;
- wsparcie;
- utrzymanie;
- rozwój;

ITIL v2 – Information Technology Infrastructure Library v2

- zarządzanie operacyjne;
- zarządzanie taktyczne;
- zarządzanie strategiczne;
- wsparcie;
- utrzymanie;

ITIL v3 – Information Technology Infrastructure Library v3

- zarządzanie operacyjne;
- projekty;
- zarządzanie taktyczne;
- zarządzanie strategiczne;
- wsparcie;
- utrzymanie;
- rozwój;

ITIL jest modelem obejmującym obecnie najszerzy zakres. Jest to również najbardziej popularny zbiór dobrych praktyk. Używany jest w 71% przypadków reorganizacji procesów IT w firmach [20].

2.3 Korzenie ITIL [6]

W latach osiemdziesiątych wiele osób pracujących dla brytyjskiego rządu oraz w służbach cywilnych poszukiwało bardziej efektywnych a zarazem mało kosztownych rozwiązań dla wymogów zarządzania technologiami informacyjnymi. Większość potrzeb wynikających z rozwoju technologii powstała za czasów rządów Margaret Thatcher i została zidentyfikowana przede wszystkim przez osoby pracujące na wyższych stanowiskach w ówczesnym rządzie, m.in: przez osoby pracujące w Central Computer and Telecommunications Agency (CCTA) - późniejszym Office of Government Commerce (OGC).

Tradycyjnie przyjmuje się, że idee, będące bazą późniejszego ITIL, pochodzą od Petera Skinnera oraz Johna Stewarta z CCAT. W ich mniemaniu, agencje rządowe wydawały zbyt duże sumy pieniędzy na IT. Podzielający to zdanie rząd Margaret Thatcher postanowił znaleźć sposób na bardziej efektywne oraz tańsze działania.

Wtedy też powstał nowy projekt, o nazwie Government Information Technology Infrastructure Management Method (GITIMM). Główną koncepcją GITIMM było, by ludzie stali się bardziej mobilni: różne departamenty oraz organizacje mogłyby kierować swoją strukturą IT w podobny sposób, co pozwoliłoby specjalistom IT na wdrożenie tych samych standardów i działań w różnych projektach. Dzięki takiej metodzie, miało nadzieję na podwyższenie wydajności oraz obniżenie kosztów.

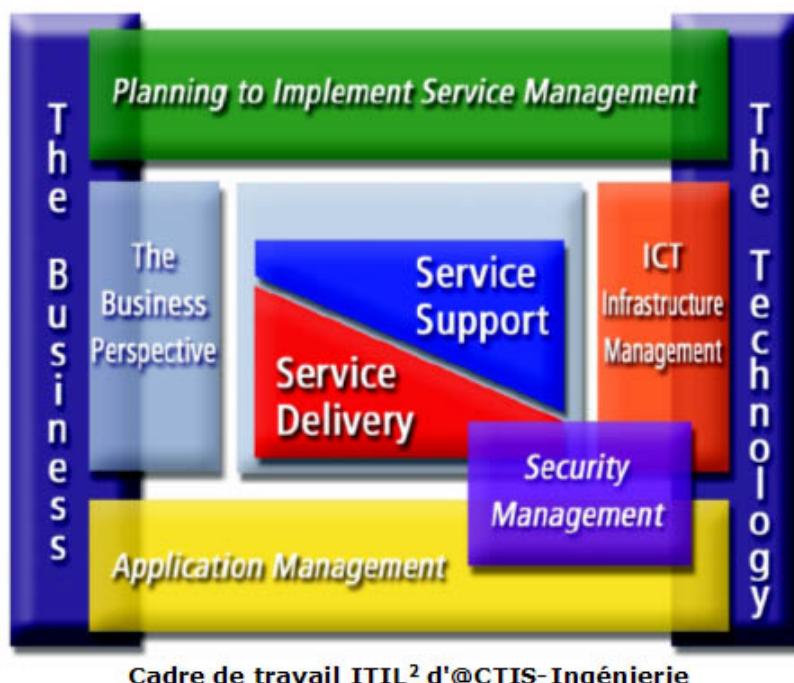
W roku 1986 John Stewart otrzymał zadanie opracowania planu stworzenia metody zwanej Government Information Technology Infrastructure Management Method. Zaczął on od tworzenia dobrego zespołu. Johnson dołączył w roku 1988 i podjął pracę nad modułami wsparcia cyklu życia oprogramowania, kosztami, pojemnością, dostępnością oraz jakością. Zespół konsultował się z wieloma przedsiębiorstwami z sektora prywatnego, co pomogło mu przedstawić pomysły dotyczące wsparcia i dostarczania usług. Między innymi sięgnął do koncepcji zarządzania usługami opisanych przez IBM. Stewart oraz jego zespół prowadzili wspomniane konsultacje w stosunkowo kreatywny oraz sprawny sposób. Poprosili kilka konkurencyjnych korporacji o dostarczenie swoich ekspertyz z różnych dziedzin. Po czym wspomniane korporacje dokonały wzajemnej edycji swoich dokumentacji. Proces, w którym przedsiębiorstwa nawzajem edytowały swoją dokumentację efektywnie przyczynił się do ograniczenia promowania własnego przedsiębiorstwa, systemów oraz technologii.

2.4 Help Desk a Service Desk [16]

Podstawowym celem Help Desk jest zarządzanie incydentami oraz dążenie do jak najszybszego ich rozwiązania. Zgłoszenia kierowane są do poszczególnych konsultantów. Ważny jest czas reakcji oraz czas realizacji. Natomiast Service Desk jest

dodatkowo interfejsem do zarządzania zmianą, licencjami, finansami, dostępnością i konfiguracją. W tym rozwiązaniu zgłoszenia są kierowane do grup, a nie do konkretnych osób. Liczy się procent reakcji oraz procent realizacji. Procent realizacji oznacza jaki odsetek zgłoszeń jest rozwiązywany w czasie krótszym niż czas określony poprzez umowę SLA³, analogicznie jest z procentem reakcji. Nie jest ważny dokładny czas realizacji, natomiast ważne jest, czy nie zostały przekroczone warunki umowy.

2.5 ITIL 2 - procesy



Rysunek 2.1: Procesy ITIL2
www.actis-ingenieure.com/ITIL.jpg

Biblioteka ITIL v2 prezentuje podejście procesowe do zarządzania usługami informatycznymi. Zarządzanie usługami informatycznymi zostało podzielone na dwa kluczowe obszary: Service Support i Service Delivery [6].

2.5.1 Service Support

Zarządzanie incydentami – Service Desk and Incident Management

Podstawowym celem procesu zarządzania incydentami jest przywrócenie normalnego poziomu usługi w jak najkrótszym czasie oraz ograniczenie wpływu na biz-

³SLA - Service Level Agreement

nes. Normalny poziom usługi jest definiowany w dokumentach SLA⁴.

Zarządzanie problemami – Problem Management

Podstawowy cel zarządzania problemami to zapobieżenie ewentualnym incydentom oraz problemom. Proces ten udostępnia rozwiązania tymczasowe lub docelowe, które mogą zostać wykorzystane przy rozwiązywaniu incydentów. Dodatkowo minimalizuje wpływ incydentów, które nie mogą być powstrzymane.

Zarządzanie zmianą – Change Management

Wszystkie zmiany muszą być upoważnione oraz udokumentowane, aby zmniejszyć ryzyko wystąpienia niezaplanowanych zdarzeń. Należy zaplanować kroki implementacji. Za wszystko odpowiedzialny jest menedżer zmian⁵ oraz CAB⁶.

Zarządzanie wydaniami – Release Management

Proces ten odpowiada za implementację zmian, w sposób efektywny, bezpieczny oraz możliwy do weryfikacji.

Zarządzanie konfiguracją – Configuration Management

Udostępnia informacje o infrastrukturze IT. Wszystkie zmiany są dokumentowane, więc dostępne są aktualne oraz historyczne dane o elementach konfiguracji⁷ w CMDB⁸.

2.5.2 Service Delivery

Zarządzanie poziomem usług – Service Level Management

Proces ten ma za zadanie utrzymywanie katalogu usług oraz osiąganie porozumienia w sprawie ich wydajności. Poziom usług jest ustalany w dokumentach SLA⁹. Zarządzanie poziomem usług jest odpowiedzialne za kontrolowanie ustalonej jakości usług. Dokumenty OLA¹⁰ określają poziom wewnętrznych usług, od których zależne są usługi biznesowe (usługi postrzegane przez klienta).

Zarządzanie dostępnością – Availability Management

Jest to proces sprawdzający, czy dostępność usług IT jest zgodna z ustalonym poziomem usług w dokumencie SLA.

⁴SLA - Service Level Agreement

⁵Change Manager

⁶CAB – Change Advisory Board

⁷element konfiguracji (ang. Configuration Item)

⁸Configuration Management Database

⁹SLA - Service Level Agreement

¹⁰OLA - Operational Level Agreement

Zarządzanie pojemnością – Capacity Management

Proces zarządzania pojemnością jest odpowiedzialny za świadczenie usług IT w sposób optymalny i efektywny kosztowo. W ramach tego procesu szacowane są przyszłe potrzeby, co umożliwia planowanie zapotrzebowania na poszczególne usługi IT.

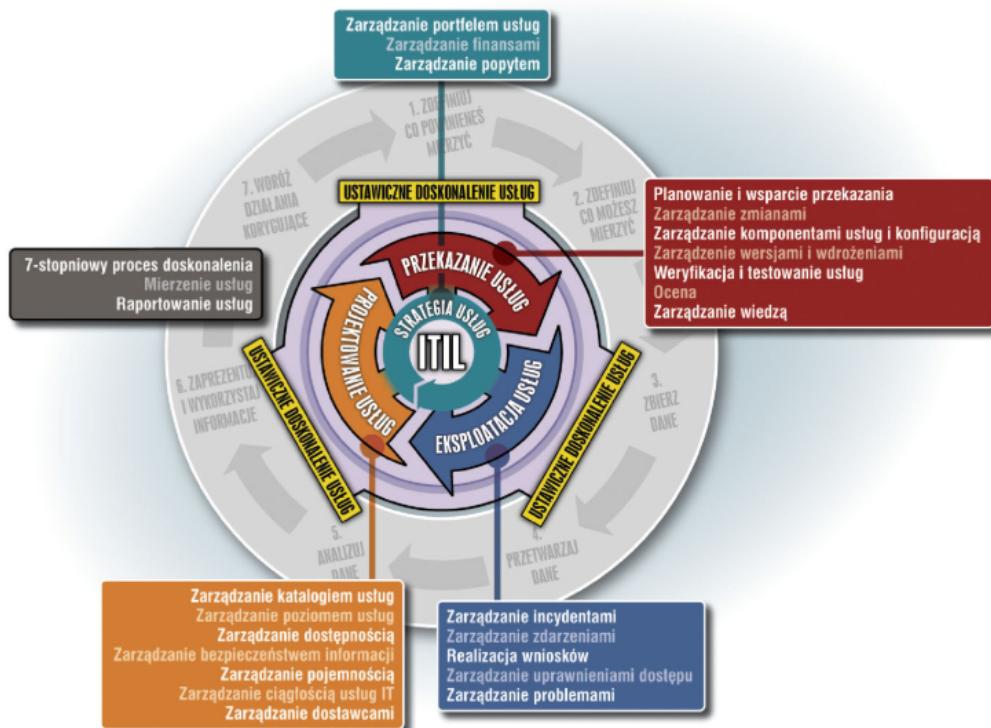
Zarządzanie ciągłością usług IT – IT Service Continuity Management

Proces ten odpowiada za zdefiniowanie i zaplanowanie środków zaradczych w przypadku wystąpienia zdarzeń katastrofalnych w skutkach.

Zarządzanie finansami usług IT – Financial Management for IT Services

Jest to proces odpowiedzialny za pobieranie opłat od klientów za korzystanie z usług. Określa również kompromis pomiędzy jakością usług, a ich kosztem, uwzględniając potrzeby klientów.

2.6 ITIL 3 - obszary



Rysunek 2.2: Obszary ITIL3

<http://itsm.itlife.pl/images/itsm/lifecycle4.jpg>

Biblioteka ITIL v3 wprowadziła model cyklu życia usługi, podzielony na pięć kluczowych obszarów: Strategia usług, Projektowanie usług, Wdrażanie usług, Eksploatacja usług oraz Ciągłe doskonalenie usług [6].

2.6.1 Strategia usług – Service Strategy [17]

Zarządzanie portfelem usług – Service Portfolio Management

Określenie strategii udostępniania usług dla klientów oraz rozwijania ofert dostawców usług.

Zarządzanie finansami – Financial Management

Zarządzanie budżetem oraz opłatami dla dostawców usług IT.

2.6.2 Projektowanie usług – Service Design [15]

Zarządzanie katalogiem usług – Service Catalogue Management

Proces ten jest odpowiedzialny za zapewnienie aktualnych informacji w katalogu usług. Dostarcza dane o usługach pozostałym procesom.

Zarządzanie poziomem usług – Service Level Management

Proces ten ustala poziom usług dla klientów oraz projektuje usługi z uwzględnieniem wcześniej ustalonego poziomu. Odpowiada również za dopilnowanie warunków zawartych w umowach OLA¹¹ oraz UC¹².

Zarządzanie ryzykiem – Risk Management

Określenie, wycena oraz kontrola ryzyka. W ramach tego procesu należy określić wartość zasobów¹³, zidentyfikować zagrożenia oraz określić wpływ zagrożeń na zasoby.

Zarządzanie pojemnością – Capacity Management

Zapewnienie, że pojemność usług IT oraz infrastruktura IT jest zdolna do dostarczenia usług IT na ustalonym poziomie, w sposób efektywny kosztowo z uwzględnieniem czasu.

¹¹OLA - Operational Level Agreement

¹²UC - Underpinning Contract

¹³zasób (ang. asset) - komputery, serwery, drukarki, macierze dyskowe, elementy aktywne oraz pasywne sieci komputerowych.

Zarządzanie dostępnością – Availability Management

Definiowanie, analizowanie, planowanie, mierzenie oraz udoskonalanie wszystkich aspektów dostępności usług IT. Jest to proces odpowiedzialny za zapewnienie tego, aby infrastruktura, procesy, narzędzia, role były odpowiednie dla ustalonych poziomów dostępności.

Zarządzanie ciągłością usług IT – IT Service Continuity Management

Zarządzanie ryzykiem, które może mieć poważny wpływ na usługi IT. Zapewnia, że dostawcy usług zawsze mogą dostarczać minimalny poziom usług, poprzez redukowanie ryzyka zdarzeń o katastrofalnych skutkach do akceptowalnego poziomu oraz planowanie przywracania usług IT.

Zarządzanie bezpieczeństwem – IT Security Management

Zapewnienie poufności, integralności, dostępności informacji w organizacji, danych oraz usług IT.

Zarządzanie zgodnością – Compliance Management

Dopilnowanie zgodności usług IT, procesów oraz systemów z polityką firmy oraz wymaganiami prawnymi.

Zarządzanie architekturą – IT Architecture Management

Utworzenie planów dla przyszłego rozwoju, przy uwzględnieniu strategii usług oraz nowych technologii.

Zarządzanie dostawcami – Supplier Management

Dopilnowanie wsparcia dla biznesu przez dostawców oraz przestrzegania kontraktów.

2.6.3 Przekazanie usług – Service Transition [18]**Zarządzanie zmianami – Change Management**

Kontrolowanie cyklu życia wszystkich zmian. Podstawowym celem jest umożliwienie przeprowadzenia korzystnych zmian z minimalnym zakłóceniem usług IT.

Zarządzanie projektami – Project Management (Transition Planning and Support)

Planowanie zasobów przy wdrożeniu najważniejszych wydań w zakresie zaplanowanych kosztów, czasu oraz jakości.

Zarządzanie wydaniami i wdrożeniami – Release and Deployment Management

Planowanie, harmonogramowanie oraz kontrolowanie przeniesienia wydań na środowiska testowe oraz produkcyjne. Podstawowy cel to zapewnienie stabilności środowisk produkcyjnych.

Weryfikacja i testowanie usług – Service Validation and Testing

Zapewnienie spełnienia oczekiwania klientów wobec wydań oraz usług. Dopilnowanie możliwości wsparcia dla nowych usług przez operacyjną część IT.

Rozwój oraz modyfikacja aplikacji – Application Development and Customization

Zapewnienie dostępności aplikacji oraz systemów dostarczających funkcjonalność dla usług IT. W zakres tego procesu wchodzi rozwijanie oraz utrzymywanie aplikacji oraz konfiguracja nabytych produktów.

Zarządzanie komponentami usług i konfiguracją – Service Asset and Configuration Management

Utrzymanie informacji o elementach IT¹⁴ wymaganych do dostarczania usług IT. Również zarządzanie relacjami pomiędzy tymi elementami.

Zarządzanie wiedzą – Knowledge Management

Gromadzenie, analizowanie, przechowywanie oraz dzielenie się wiedzą i informacją w ramach całej organizacji. Podstawowym celem jest zwiększenie efektywności poprzez ograniczenie sytuacji ponownego odkrywania wiedzy.

2.6.4 Eksploatacja usług – Service Operation [16]**Zarządzanie zdarzeniami – Event Management**

Proces odpowiedzialny za filtrację oraz kategoryzację zdarzeń zachodzących w infrastrukturze IT oraz podjęcie odpowiedniej akcji.

Zarządzanie incydentami – Incident Management

Zarządzanie cyklem życia incydentów. Podstawowym celem jest dokonanie naprawy usługi IT w jak najkrótszym czasie.

¹⁴Element IT (ang. Configuration Item) – komputer, serwer, drukarka itp.

Realizacja wniosków – Request Fulfilment

Obsługiwanie standardowych zmian typu: zmiana hasła oraz instalacja oprogramowania.

Zarządzanie uprawnieniami dostępu – Access Management

Nadawanie uprawnień uprzywilejowanym użytkownikom do korzystania z usługi oraz uniemożliwienie korzystania z usługi klientom, którzy tych uprawnień nie posiadają.

Zarządzanie problemami – Problem Management

Zarządzanie problemami jest odpowiedzialne za ograniczenie pojawiania się nowych incydentów oraz minimalizację wpływu już istniejących.

Zarządzanie operacjami – IT Operations Management

Monitorowanie oraz kontrola usług IT oraz infrastruktury IT. Planowanie zadań, zarządzanie kopiami zapasowymi.

Zarządzanie instalacjami IT – IT Facilities Management

Zarządzanie fizycznymi środowiskami, w których zlokalizowana jest infrastruktura IT. Zarządzanie energią, ogrzewaniem oraz monitorowaniem środowiska.

2.6.5 Ustawiczne doskonalenie usług – Continual Service Improvement [14]**Ocena usług – Service Evaluation**

Ocena jakości usług. Identyfikacja obszarów, w których nie są osiągane założone poziomy usług, przeprowadzanie rozmów z biznesem, w celu upewnienia się, czy uzgodnione poziomy usług są zgodne z potrzebami biznesu.

Ocena procesów – Process Evaluation

Ocena procesów pod względem fundamentalnym. Identyfikacja obszarów, w których nie są osiągane metryki procesów, przeprowadzanie audytów, ocena dojrzałości oraz recenzje procesów.

Definicja inicjatyw udoskonalania – Definition of Improvement Initiatives

Na podstawie oceny procesów oraz usług, zdefiniowanie inicjatyw, mających na celu udoskonalenie usług oraz procesów.

2.7 Szczegółowe omówienie wybranych procesów

W tej sekcji zostaną omówione wybrane procesy ITIL: zarządzania incydentami, zarządzania problemami, zarządzania zdarzeniami oraz zarządzania zleceniami. Autor niniejszej pracy zdecydował się na wybranie procesu zarządzania incydentami, ponieważ jest to proces, od którego rozpoczyna się wdrażanie systemu Service Desk. Pozostałe procesy w dużym stopniu bazują na danych dotyczących incydentów. Następnie, zgodnie z przyjętą praktyką, wdraża się proces zarządzania problemami. Umożliwia to zmniejszenie liczby incydentów oraz szybsze ich rozwiązywanie. Następne procesy to zarządzanie zmianą oraz zarządzanie konfiguracją. W tej pracy zostały one pominięte, ponieważ są one tematem pracy dyplomowej inżynierskiej Adriana Wiśniewskiego, realizowanej równolegle. Autor zdecydował się na wybranie dwóch kolejnych procesów z obszaru Eksplotacji usług: zarządzania zdarzeniami oraz zarządzania zleceniami, które są silnie związane z funkcją Service Desk. Zarządzanie zdarzeniami umożliwia wygenerowanie nowego incydentu w odpowiedzi na zaistnienie zdarzenia w infrastrukturze IT, natomiast zarządzanie zleceniami jest odpowiedzialne za realizowanie standardowych zmian. Jest to proces, który wyodrębnia zarządzanie standardowymi zmianami z procesu zarządzania zmianami ze względu na zmniejszenie kosztów.

2.7.1 Zarządzanie incydentami [16]

Incydent jest to niezaplanowane przerwanie działania usługi IT lub ograniczenie jakości usługi IT.

Cel

Podstawowym celem procesu zarządzania incydentami jest przywrócenie normalnego poziomu usługi w jak najkrótszym czasie oraz ograniczenie wpływu na biznes. Normalny poziom usługi jest definiowany w dokumentach SLA¹⁵.

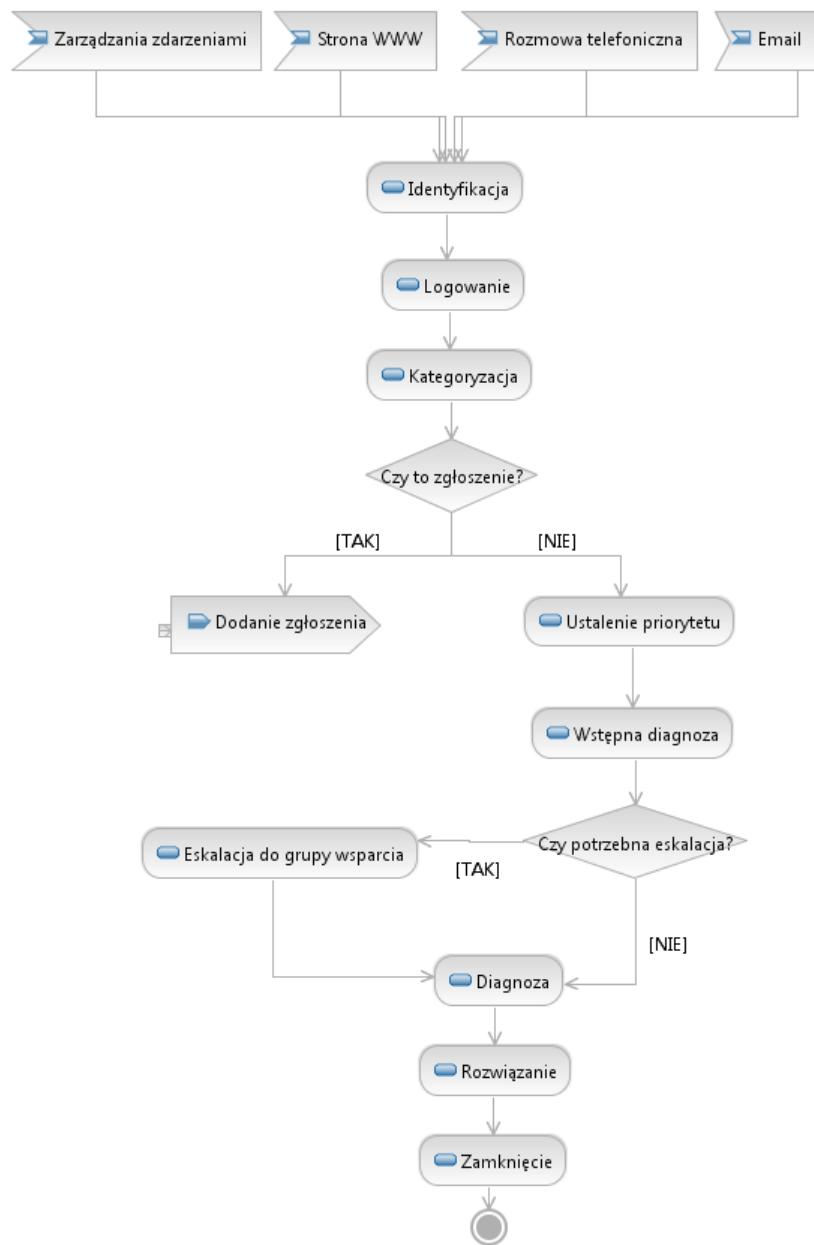
Identyfikacja

Identyfikacja jest to ustalenie wskazujące na wystąpienie incydentu. Z biznesowego punktu widzenia nie można czekać, aż użytkownik zgłosi incydent. Wszystkie kluczowe elementy powinny być monitorowane, aby jak najszybciej wykryć wystąpienie incydentu. W idealnym przypadku wszystkie incydenty powinny zostać rozwiązane zanim użytkownicy zostaną ograniczeni w korzystaniu z usługi.

Logowanie

Wszystkie incydenty muszą zostać zapisane oraz oznakowane stemplem czasu.

¹⁵ SLA - Service Level Agreement



Rysunek 2.3: Zarządzanie incydentami - proces

Każdy incydent musi posiadać następujące dane:

- unikalny identyfikator;
- kategorię;
- wpływ;
- pilność;

- priorytet;
- datę zgłoszenia;
- serwisanta oraz grupę wsparcia;
- źródło: formularz WWW, e-mail, telefon, automatycznie poprzez zarządzanie zdarzeniami;
- status;
- rozwiązanie;
- datę rozwiązania;
- datę zamknięcia;

Kategoryzacja

Każdy incydent powinien mieć przypisaną kategorię. W dalszych etapach będzie to ważne w ustaleniu, jakie zdarzenia występują najczęściej. Kategorie powinny tworzyć wielopoziomową hierarchię (zwykle wystarczają cztery poziomy).

Przykładowa hierarchia:

- Sprzęt
 - Serwer
 - * Karta pamięci
 - * Pamięć trwała
 - Oprogramowanie
 - Aplikacja
 - * Aplikacja finansowa
 - * CRM
 - Przeglądarka internetowa
 - * Firefox
 - * IE

Każda organizacja powinna ustalić hierarchię odpowiednią do jej potrzeb.

Priorytet

Każdy incydent powinien mieć określony priorytet. Priorytet powinien zostać ustalony na podstawie pilności, która uwzględnia potrzeby biznesu oraz na podstawie wpływu na biznes. Wpływ jest zwykle określany na podstawie liczby użytkowników, którzy są połączeni z danym incydentem.

Efektywnym sposobem ustalenia priorytetu jest stworzenie macierzy odwzorowującej wpływ oraz pilność na priorytet **2.1**.

Tabela 2.1: Kodowanie priorytetu

			Wpływ	
		Wysoki	Średni	Niski
Pilność	Wysoka	1	2	3
	Średnia	2	3	4
	Niska	3	4	5

Każdy priorytet powinien określać czas rozwiązania incydentu **2.2**.

Tabela 2.2: Czas rozwiązania

Priorytet	Czas rozwiązania [h]
1	1
2	8
3	24
4	48
5	96

Eskalacja

W sytuacji, gdy incydent nie może zostać rozwiązany przez pracowników Service Desk, powinien zostać jak najszybciej eskalowany do specjalistycznych grup wsparcia. Analogicznie, jeżeli druga linia nie jest w stanie w ustalonym czasie rozwiązać incydentu, to powinien on zostać eskalowany do trzeciej linii. Teoretycznie eskalacja może mieć dowolną liczbę poziomów, ale zwykle powinna się ograniczać do trzech. Należy dodać, że grupy wsparcia mogą składać się z pracowników firmy lub przedstawicieli firm zewnętrznych, przykładowo dostawców oprogramowania lub sprzętu.

Rozwiązanie

W momencie, gdy zostanie znalezione potencjalne rozwiązanie, powinno ono zostać zastosowane i przetestowane. To jakie akcje zostaną wykonane oraz kto będzie zaangażowany w ich wykonanie zależy od incydentu. W jednym przypadku może zdarzyć się, że użytkownik zostanie poproszony o wprowadzenie zmiany

w odrębnie swojego środowiska, w innym serwisant może dokonać rozwiązania w sposób centralny, przykładowo poprzez zrestartowanie serwera. W niektórych przypadkach zewnętrzni dostawcy będą poproszeni o rozwiązanie pozwalające na usunięcie przyczyny, a przynajmniej skutku błędu.

Zamknięcie

Przy zamknięciu incydentu pracownicy Service Desk powinni sprawdzić, czy użytkownik jest zadowolony z rozwiązania. Dodatkowo należy sprawdzić, czy dane na temat incydentu zostały wypełnione poprawnie oraz czy są kompletne.

Powiązane procesy

Zarządzanie zdarzeniami – Zdarzenia w infrastrukturze IT mogą spowodować wygenerowanie incydentu.

Zarządzanie problemami – Incydenty są skutkiem istnienia problemów.

Zarządzanie konfiguracją – Udostępnia dane potrzebne do zidentyfikowania incydentów.

Zarządzanie poziomem usług – Rozwiązywanie incydentów w określonym czasie jest ważnym elementem udostępniania usług na ustalonym poziomie.

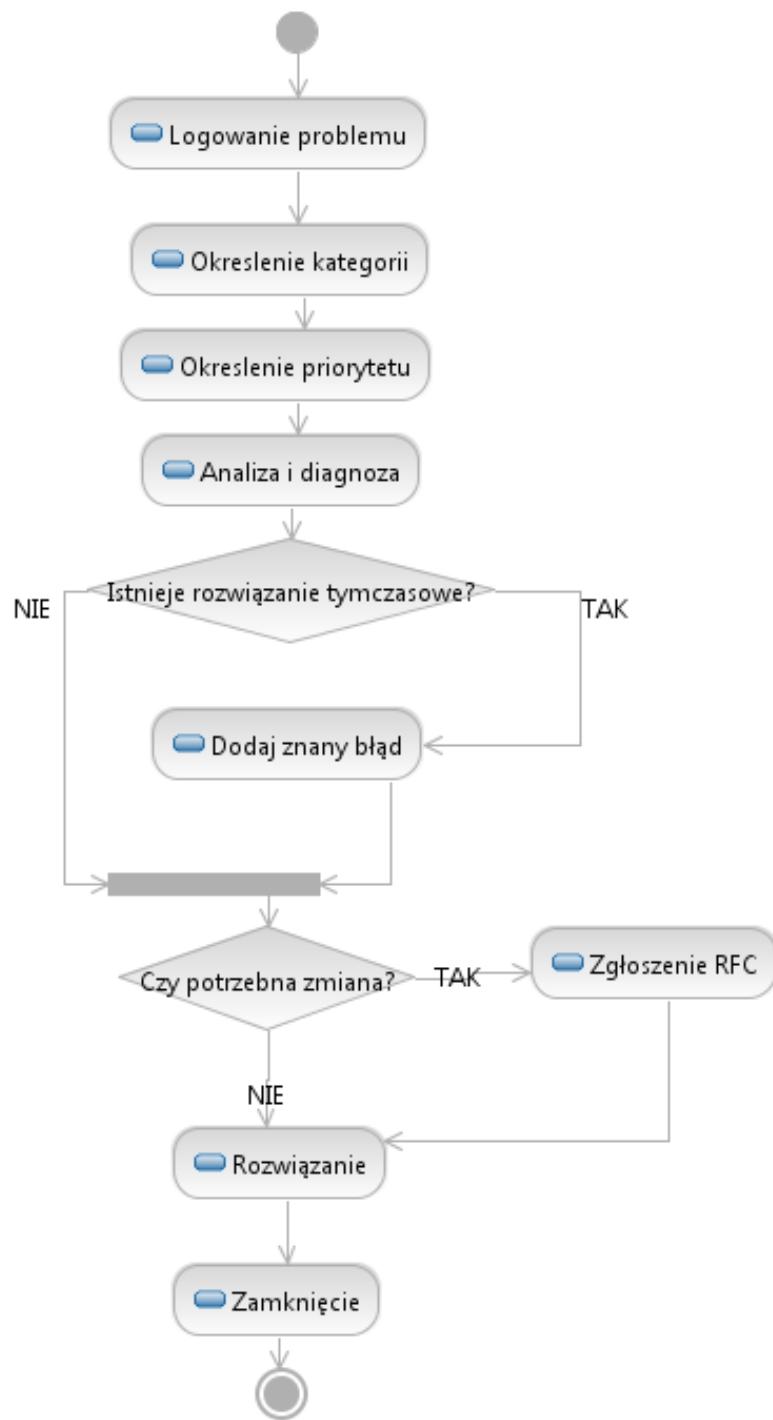
Zarządzanie katalogiem usług – Dostarcza informacje o usługach przypisanych do klienta.

2.7.2 Zarządzanie problemami [16]

Problem jest definiowany jako przyczyna wystąpienia jednego lub wielu incydentów[16].

Cel

Podstawowym celem zarządzania problemami jest zapobieżenie ewentualnym incydentom oraz problemom. Dodatkowo, zminimalizowanie wpływu incydentów, które nie mogą być powstrzymane. Zarządzanie problemami ma wpływ na zwiększenie jakości usług IT. Przy rozwiązywaniu nowych incydentów można wykorzystać doświadczenie nabycie w przeszłości, co wpłynie na przyspieszenie momentu ich rozwiązania. Dzięki temu zwiększa się dostępność krytycznych systemów dla biznesu.



Rysunek 2.4: Zarządzanie problemami - proces

Logowanie

Wszystkie problemy powinny zawierać następujące dane:

- temat;
- opis;
- użytkownik, który zgłosił problem;
- data i godzina dodania;
- priorytet oraz kategoria;
- powiązane incydenty;

Analiza i diagnoza

W trakcie analizy problemu należy określić główną przyczynę jego wystąpienia. Pomocnym narzędziem jest CMS¹⁶, system pomagający oszacować wielkość wpływu. Należy również skorzystać z KEDB¹⁷ w celu przeszukania podobnych problemów.

Rozwiążanie tymczasowe

W niektórych przypadkach jesteśmy w stanie znaleźć tymczasowe rozwiązanie dla problemu. Takie rozwiązanie może zostać wykorzystane przy zarządzaniu incydentami.

Baza znanych błędów

Po ukończeniu diagnozy, gdy znana jest przyczyna problemu oraz udało się znaleźć rozwiązanie tymczasowe, należy dodać nowy znany błąd¹⁸. W celu uniknięcia duplikatów, jedyną osobą, która może to zrobić powinien być menedżer problemów. Znane błędy tworzą KEDB¹⁹. Głównym jej celem jest przechowywanie wiedzy na temat poprzednio rozwiązanych incydentów oraz problemów. Umożliwia to szybsze działanie w przyszłości oraz wdrożenie rozwiązania dla przypadków, które się powtarzają. Każdy znany błąd powinien zawierać informacje o awarii oraz o symptomach jakie się ujawniły, rozwiązanie stałe oraz tymczasowe.

Powiązane procesy

Zarządzanie incydentami – problem jest przyczyną powstania wielu incydentów.

Zarządzanie zmianami – rozwiązanie problemów jest związane z wprowadzeniem zmiany.

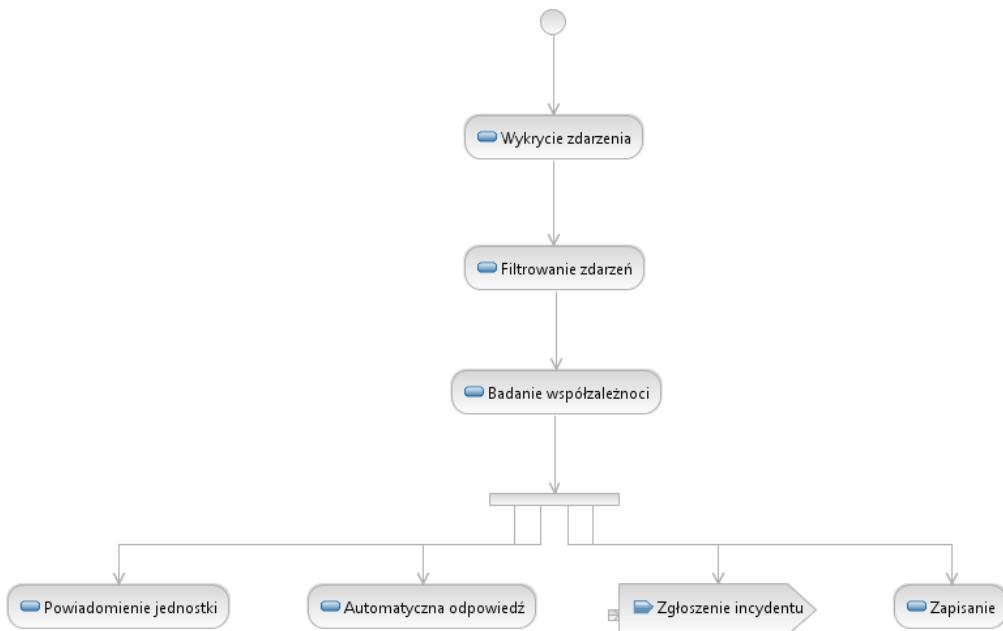
¹⁶CMS - Configuration Management System

¹⁷KEDB - Known Error Database

¹⁸znany błąd – (ang. known error)

¹⁹KEDB – Known Error Database

2.7.3 Zarządzanie zdarzeniami [16]



Rysunek 2.5: Zarządzanie zdarzeniami - proces

Zdarzenie jest to dowolne wystąpienie związane z zarządzaniem infrastrukturą IT lub z funkcjonowaniem usługi IT.

Cel

Zarządzanie zdarzeniami ma na celu wykrywanie zdarzeń oraz podejmowanie odpowiedniej akcji. Jest to również punkt startowy dla innych procesów: zarządzania incydentami, problemami.

Wystąpienie zdarzenia

Zdarzenia występują ciągle, ale nie wszystkie z nich są wykrywane oraz rejestrowane. Każda organizacja musi podjąć decyzję co i w jakim stopniu ma być monitorowane.

Filtrowanie zdarzeń

Na tym etapie należy zdecydować, czy zdarzenie powinno zostać przekazane dalej czy zignorowane. Zignorowanie zdarzenia oznacza, że pozostanie w logach, ale nie zostanie podjęta żadna akcja.

Ranga

Zdarzenia są dzielone ze względu na rangę. ITIL zaleca uwzględnienie następujących rang:

Informacyjne – są to zdarzenia, które nie wymagają podjęcia żadnej akcji. Mogą zostać wykorzystane do wygenerowania statystyk, np. o liczbie logowań do systemu.

Ostrzeżenie – są to zdarzenia, które oznaczają prawdopodobieństwo naruszenia ustalonych warunków. Wykorzystywane są do poinformowania odpowiedniej jednostki w celu sprawdzenia sytuacji oraz podjęcia działania, aby uniknąć awarii. Przykładem takiego zdarzenia jest zbliżenie się do limitu wykorzystania pamięci.

Alarm – jest to zdarzenie oznaczające nieprawidłową pracę urządzenia, usługi.

Współzależność zdarzeń

Dla ważnych zdarzeń należy sprawdzić jak bardzo są one istotne oraz jaką akcję należy podjąć. Dokonuje się tego poprzez sprawdzenie, czy dane zdarzenie spełnia szereg reguł. Badana jest wielkość wpływu zdarzenia. Przykładowe sprawdzane czynniki:

- liczba podobnych zdarzeń;
- liczba elementów generujących podobne zdarzenia;
- czy został przekroczyony limit?
- czy zdarzenie jest ostrzeżeniem?
- czy potrzebne są dodatkowe dane do podjęcia decyzji?

Wyzwalacz

Jeśli w poprzednim etapie zostanie wykryte zdarzenie, należy podjąć odpowiednią akcję. Przykładowe akcje:

- wygenerowanie incydentu;
- wygenerowanie RFC²⁰;
- wykonanie skryptu;
- powiadomienie odpowiedniej jednostki;

²⁰RFC - Request For Change

Powiązane procesy

Zarządzanie incydentami – zdarzenie, może wygenerować incydent.

2.7.4 Zarządzanie zleceniami [16]

Zlecenie²¹ jest to pewien rodzaj zmiany, która niesie ze sobą małe ryzyko, często się pojawia i jest mało kosztowna. Są to standardowe zmiany w organizacji. Przykładem jest zmiana hasła lub instalacja oprogramowania.

Proces ten powstał, aby w sposób efektywny zarządzać takim rodzajem zmian. Obsługiwanie ich poprzez proces zarządzania zmianami byłoby skrajnie nieefektywne. Czasami zlecenia obsługiwane są w procesie zarządzania incydentami, ale ITIL v3 nie zaleca takiej praktyki, ponieważ istnieje zasadnicza różnica między incydentem a zleceniem. Incydent jest czymś z definicji niezaplanowanym, a zlecenia powinny być planowane.

Cel

Celem zarządzania zleceniami jest udostępnienie klientom sposobu zgłaszania i otrzymywania usług, dla których istnieje dobrze zdefiniowany proces.

²¹Zlecenie - ang. Service Request

Rozdział 3

Firma

Rozdział ten przedstawia opisanie wykroowanej firmy dostarczającej usługi IT. Na tym przykładzie zostało zaprezentowane wsparcie zrealizowanego fragmentu systemu dla jej działalności.

3.1 Opis ogólny

RedHost S.A. jest firmą świadczącą profesjonalne usługi IT w zakresie hostingu. Przedsiębiorstwo posiada własne centrum danych (data center) znajdujące się w Warszawie, które składa się z czterech pomieszczeń pełniących role serwerowni, mieszczących łącznie ponad dwieście szaf na serwery (server rack). Firma świadczy kompleksowe usługi polegające na odpłatnym udostępnianiu zasobów: od wynajmowania miejsc w serwerowniach przez udostępnianie wirtualnych systemów operacyjnych po dzierżawienie wydzielonego miejsca na dyskach twardych, macierzy dyskowych i serwerów. Klientami firmy są przedsiębiorstwa każdej wielkości oraz osoby prywatne, które dzięki wynajmowi nie muszą ponosić kosztów utrzymania skomplikowanej infrastruktury, która zapewnia niezawodność i szybkość serwerów.

3.2 Misja

Świadczenie profesjonalnych usług hostingowych, dzięki którym klienci będą mogli z powodzeniem realizować i rozwijać swoje przedsiębiorstwa.

3.3 Cele

- Osiągnięcie i stabilizacja co najmniej 10 pozycji w rankingu największych firm hostingowych w Polsce.

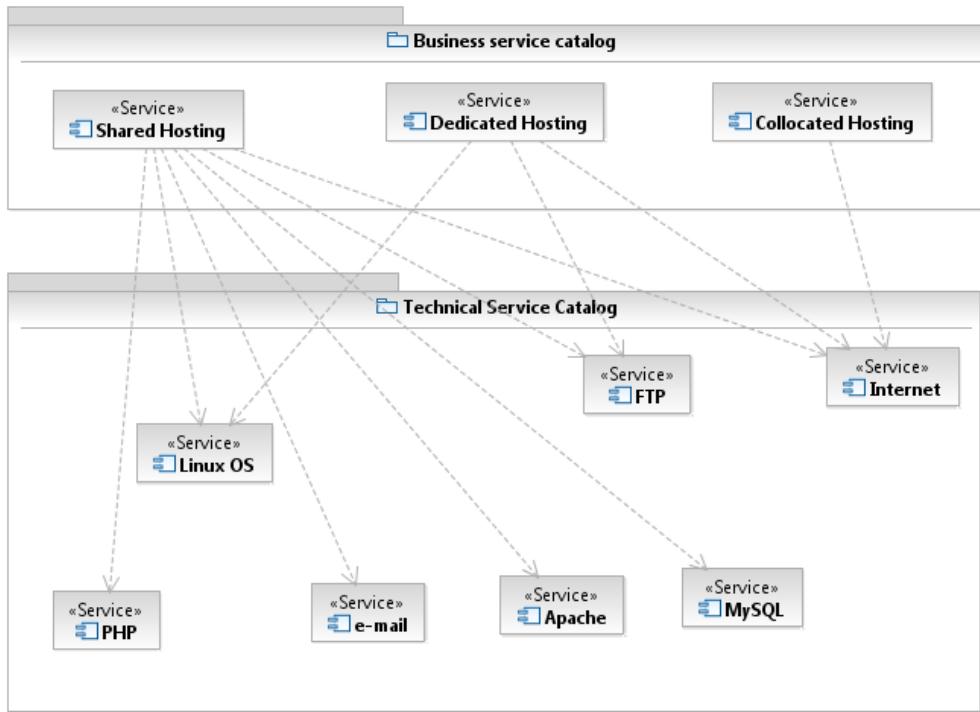
- Nawiązanie kontaktów biznesowych z właścicielami popularnych polskich portali internetowych.
- Wejście na rynek europejski.
- Ciągła modernizacja i podnoszenie jakości świadczonych usług.

3.4 Profil firmy

W epoce informacyjnej, w której obecnie żyjemy, praca wielu przedsiębiorstw jest wspomagana lub nawet w całości zależy od systemów informatycznych przechowujących i przetwarzających informacje. Przerwy w świadczeniu usług przez te systemy spowodowane awariami zaburzają operatywność przedsiębiorstwa i powodują wymierne straty. Aby zwiększyć niezawodność i zminimalizować ryzyko awarii stosuje się redundancję źródeł zasilania, łączy internetowych, dysków twardych, serwerów i innych elementów niezbędnych do działania systemu informatycznego. Jednak takie postępowanie prowadzi do znacznego zwiększenia złożoności infrastruktury informatycznej, a w konsekwencji zwiększenia kosztów utrzymania systemu i wymaga zatrudnienia dodatkowych, odpowiednio wykwalifikowanych pracowników. Małe i średnie przedsiębiorstwa nie są w stanie pokryć dodatkowych kosztów związanych z utrzymaniem takiego systemu, natomiast duże przedsiębiorstwa stawiają wysokie wymagania dotyczące jakości i bezpieczeństwa. Ponadto większość przedsiębiorstw ma wystarczająco wiele obowiązków i po prostu nie ma zamiaru zajmować się problemami informatycznymi. Dlatego istnieje potrzeba zlecenia opieki nad systemem informatycznym przedsiębiorstwa zewnętrznej firmie, która dzięki swojej specjalizacji jest w stanie zapewnić odpowiednie zaplecze techniczne i wyszkolonych fachowców. W dodatku firma ta jest w stanie rozłożyć koszt utrzymania całej infrastruktury na wielu klientów, co pozwala na obniżenie cen świadczonych usług. Na tym właśnie polega hosting, którym zajmuje się firma RedHost S.A. Usługi są świadczone także osobom prywatnym, które chcą zamieścić swoją stronę internetową w sieci.

3.5 Usługi

W zależności od potrzeb i zaawansowania technicznego klienta hosting może przyjmować różne formy. Przedsiębiorstwa, które chcą posiadać niczym nieskrępowaną możliwość konfiguracji swojego systemu, będą zainteresowane wynajęciem miejsca w serwerowni, w którym będą mogły ustawić swój sprzęt i same będą nim zarządzać. Z drugiej strony duża część przedsiębiorstw zamierza delegować wszystkie obowiązki związane z utrzymaniem systemu. W związku z tym firma RedHost S.A świadczy zróżnicowane i kompleksowe usługi hostingowe.



Rysunek 3.1: Katalog usług firmy Red Host S.A.

Hosting kolokacyjny

Polega na ulokowaniu sprzętu należącego do klienta w serwerowni firmy świadczącej usługi hostingowe. Klient uzyskuje w ten sposób dostęp do szybkiego łącza internetowego, awaryjnych źródeł zasilania, systemu chłodzenia i innych elementów infrastruktury serwerowni ponosząc niewielkie koszty. Zachowuje przy tym możliwość pełnej administracji nad konfiguracją sprzętową i oprogramowaniem.

Hosting kolokacyjny wiąże się z potrzebą zapewnienia klientowi fizycznego dostępu do serwerowni. Ze względów bezpieczeństwa wymaga to stosowania dodatkowych zabezpieczeń uniemożliwiających niepowołany dostęp do serwerów. Najczęściej stosowanym rozwiązaniem tego problemu jest umieszczanie maszyn w specjalnych szafach z zamkiem na kartę magnetyczną lub wymagającym identyfikacji biometrycznej. Dodatkowo każda serwerownia jest pod stałym nadzorem systemu monitorującego i kamer wideo.

Ten typ hostingu jest najlepszym wyborem, jeżeli klient ze względów bezpieczeństwa lub wydajności chce ulokować kopie swojego systemu w różnych lokalizacjach geograficznych.

Hosting dedykowany

Inaczej zwany hostingiem zarządzanym, jest usługą w której firma hostingowa wynajmuje klientowi należące do niej serwery w całości. Klient posiada pełną kontrolę nad wynajętymi maszynami wliczając wybór podzespołów i systemu operacyjnego. Hosting dedykowany może niejednokrotnie być tańszy i bardziej opłacalny od kupna własnego sprzętu.

Firmy hostingowe posiadają specjalne umowy z dostawcami oprogramowania, które umożliwiają im na podstawie miesięcznych opłat udostępnianie licencji zakupionego oprogramowania własnym klientom. Dzięki temu firma hostingowa może zainstalować komercyjne systemy operacyjne i oprogramowanie na życzenie klienta, który nie musi trudzić się ich kupnem. Wiąże się to z dodatkowymi opłataми za usługę hostingową, jednak klient może znacznie zaoszczędzić na kosztach licencji.

Hosting dedykowany może wiązać się z dodatkową usługą administracji serwerem. Wyróżnia się cztery poziomy wsparcia:

- Brak zarządzania - Brak ingerencji ze strony usługodawcy. Klient sam jest odpowiedzialny za zarządzanie serwerem.
- Samodzielnego zarządzanie - Usługodawca monitoruje stan serwera i przeprowadza pewne prace konserwacyjne. Klient wykonuje większość czynności administracyjnych.
- Częściowe zarządzanie - Oprócz monitoringu i prac konserwacyjnych, usługodawca w pewnym stopniu zarządza serwerem. Klient może wykonywać określone czynności administracyjne.
- Pełne zarządzanie - Zawiera monitoring, obsługę restartów maszyny, instalację poprawek i nowych wersji oprogramowania oraz systemów operacyjnych. Klient nie wykonuje żadnych czynności administracyjnych.

Ten typ hostingu nie wymaga zapewnienia klientowi fizycznego dostępu do serwerów. Klient zarządza serwerem przez internet najczęściej przy użyciu protokołu SSH.

Wirtualny hosting dedykowany

Dzięki technologii wirtualizacji możliwe jest podzielenie zasobów jednego fizycznego serwera i jednoczesne uruchomienie na nim wielu różnych systemów operacyjnych, tak aby każdy z tych systemów zachowywał się jak niezależna maszyna. Wirtualny hosting dedykowany jest odmianą hostingu dedykowanego, w której klient zamiast wynajmować cały serwer, wynajmuje jedną z takich maszyn. Klient

nadal zachowuje niezależność w administrowaniu swoim serwerem przy dużo niższej cenie. Ze względu na liczebność maszyn wirtualnych znajdujących się na jednym fizycznym serwerze, każda z tych maszyn ma ograniczoną ilość zasobów, takich jak czas procesora, ilość pamięci RAM, czy miejsce na dysku.

Hosting współdzielony

Typ hostingu popularny wśród klientów będących małymi przedsiębiorstwami lub osobami prywatnymi, które posiadają dużą ilość wiedzy technicznej. W usłudze tej klienci wykupują konta użytkownika systemu operacyjnego działającego na serwerze, stąd też popularną nazwą usługi jest hosting powłoki (shell hosting). Klienci mogą korzystać z zainstalowanych na serwerze programów, którymi zwykle są: serwer www, serwer poczty e-mail, repozytoria, bazy danych, kompilatory i interpretery różnych języków programowania. Mogą także instalować dodatkowe oprogramowanie, ale jedynie w ograniczonym zakresie. Z uwagi na to, że na jednej maszynie pracuje wielu użytkowników jednocześnie, każdy z nich ma z góry ustalone limity na zajętość dysku twardego, użycie pamięci ram, czasu procesora i łącza internetowego. Mimo pewnych ograniczeń, hosting ten daje znacznie większe możliwości od standardowego hostingu stron internetowych i przy tym jest dużo tańszy od hostingu dedykowanego.

Współdzielony hosting stron internetowych

Jest to usługa, w której wiele stron internetowych jest udostępnianych przez jeden serwer. Każda z tych stron znajduje się w wydzielonym obszarze dysku twardego, do którego klient ma dostęp przez protokół FTP lub specjalny sieciowy panel administracyjny. Klient nie musi posiadać praktycznie żadnej wiedzy technicznej, żeby korzystać z tej usługi, ponieważ aby zamieścić swoją stronę internetową na serwerze, wystarczy skopiować pliki do odpowiedniego katalogu. Tak jak w przypadku zwykłego hostingu współdzielonego, tak i w tym wypadku usługodawca narzuca limity na wykorzystanie zasobów serwera, aby móc zagwarantować wszystkim klientom ciągłość świadczenia usługi. Wielkość limitów jest zależna od pakietu wykupionego przez klienta. Jest to najbardziej ekonomiczna opcja dla małych przedsiębiorstw i osób prywatnych, chcących jedynie umieścić swoją stronę internetową w sieci.

Darmowy hosting stron internetowych

Odmiana współdzielonego hostingu stron internetowych, w której klient nie płaci nic za usługę, jednak są na niego narzucone bardzo rygorystyczne ograniczenia. Usługa tworzona w celach marketingowych, aby niezdecydowani klienci mogli wypróbować działanie hostingu współdzielonego, a następnie wraz z rozwojem ich stron internetowych wykupili pakiet hostingu współdzielonego.

Hosting pośredni (Reseller hosting)

Jest to usługa, w której klient wynajmuje fizyczny lub wirtualny serwer oraz uzyskuje prawo do świadczenia przy jego użyciu usług hostingowych. Typowymi klientami tej usługi są firmy tworzące portale internetowe posiadające dodatkową usługę hostingu lub przedsiębiorcy chcący założyć i rozkręcić własną firmę hostingową. Aby pośrednik mógł stać się konkurencyjny względem firm świadczących usługi hostingowe z pierwszej ręki, musi prowadzić skuteczną kampanię reklamową.

3.6 Organizacja firmy

Dział sprzedaży

Obowiązki:

- Prowadzenie różnorodnych kontaktów z klientami w celu sprzedaży konkretnego produktu.

Dział zakupów

Obowiązki:

- Nabywanie sprzętu i oprogramowania.
- Zakup usług zewnętrznych.

Dział marketingu

Obowiązki:

- Zarządzanie kampaniami marketingowymi.
- Promocje dla klientów.
- Zarządzanie reklamą.

Dział obsługi klienta

Obowiązki:

- Zarządzanie kontaktami z klientem:
 - skargi klienta;
 - pytania;

Dział prawny

Obowiązki:

- Reprezentacja firmy w sądzie.
- Sporządzanie umów.

Dział zarządzania kadrami

Obowiązki:

- Zarządzanie kadrami.
- Rekrutacja.
- Urlopy.
- Szkolenia.

Dział finansowy

Obowiązki:

- Zarządzanie finansami.
- Budżet.

Dział księgowy

Obowiązki:

- Faktury.

Dział zarządzania ryzykiem

Obowiązki:

- Zarządzanie ryzykiem operacyjnym.

Dział administracji bazami danych

Obowiązki:

- Tworzenie baz danych.
- Zarządzanie logiczną i fizyczną strukturą bazy danych.
- Zarządzanie użytkownikami.

- Zarządzanie uprawnieniami.
- Zarządzanie kopiami zapasowymi.

Dział administracji siecią

Obowiązki:

- Konfiguracja sieci.

Dział administracji systemów operacyjnych i serwerów

Obowiązki:

- Zarządzanie użytkownikami.
- Zarządzanie uprawnieniami.
- Instalacja systemów operacyjnych i maszyn wirtualnych.
- Monitoring systemów.
- Zarządzanie serwerami.
- Utrzymanie serwerów.
- Monitoring serwerów.

Dział administracji poczty elektronicznej

Obowiązki:

- Zarządzanie serwerami poczty.
- Zarządzanie kontami.
- Zarządzanie aliasami.

Dział programistyczny

Obowiązki:

- Projektowanie stron WWW.

3.7 Kooperanci

Dostawca internetu

Firma powinna posiadać co najmniej dwóch dostawców internetu, korzystających z różnych fizycznych łączy. W przypadku, gdy powstaną problemy z jednym z dostawców istnieje możliwość zmiany na drugiego.

Dostawca sprzętu

Firma powinna posiadać kontakty z dostawcami sprzętu, aby zapewnić możliwość nabycia nowych jednostek, naprawy lub wymiany wadliwych komponentów.

Dostawca oprogramowania

Oprogramowanie jest wykorzystywane na co dzień przez pracowników firmy, ale również na życzenie klienta firmowego istnieje możliwość instalacji dowolnego oprogramowania, oczywiście za odpowiednią opłatą. W bardziej skomplikowanych przypadkach należy złożyć RFC¹.

Dostawca domen

Ze względu na pełnione usługi firma powinna posiadać kontakt z dostawcami domen, aby w każdym momencie mieć możliwość zakupu ich produktów.

Dostawca energii

Analogicznie jak w przypadku dostawców internetu, potrzebnych jest co najmniej dwóch dostawców energii, którzy dostarczają ją z różnych źródeł lub kierunków.

Dostawca certyfikatów SSL

Ze względu na pełnione usługi firma powinna posiadać kontakt z dostawcami certyfikatów w celu nabycia ich produktów.

3.8 Klienci

Osoba fizyczna

Klient z tej kategorii ma ograniczone możliwości. Nie jest w stanie korzystać ze wszystkich usług oferowanych przez firmę. Ma jedynie możliwość korzystania z współdzielonego hostingu stron internetowych oraz darmowego.

Firma

Klient z tej kategorii ma możliwość korzystania ze wszystkich oferowanych usług.

¹RFC - Request For Change

3.9 Obsługa klienta

Obsługa klienta zewnętrznego

W przypadku zaistnienia sytuacji utrudniającej korzystanie z usług, każdy klient firmy ma możliwość dodawania incydentu. Można to zrobić poprzez stronę WWW, drogą mailową lub telefonicznie.

Każdy klient ma również możliwość dodawania zleceń, aby dokonać standardej zmiany w świadczonych usługach. Sposób dodawania jest taki sam, jak w przypadku incydentu.

Kategorie incydentów

- Hardware
 - Zasilanie
 - Internet
- Software
 - e-mail
 - FTP
 - Baza danych
 - * MySQL
 - * PostgreSQL
 - Certyfikat

Kategorie zleceń

- Software
 - e-mail
 - * Utworzenie konta
 - * Utworzenie aliasu
 - FTP
 - * Utworzenie konta
 - Certyfikat
 - * Utworzenie certyfikatu
 - Aplikacja
 - * Instalacja oprogramowania
 - * Usunięcie oprogramowania
 - * Aktualizacja oprogramowania

Obsługa klienta wewnętrznego

W przypadku sytuacji utrudniającej pracę, każdy pracownik firmy ma możliwość dodawania incydentu.

W celu dokonania standardowej zmiany pracownik firmy ma możliwość dodawania zleceń.

Kategorie incydentów

- Hardware
 - Stacja robocza
 - Internet
 - Drukarka
- Software
 - e-mail
 - aplikacja
 - * aplikacja finansowa
 - * aplikacja marketingowa

Kategorie zleceń

- Hardware
 - Stacja robocza
 - * Wymiana stacji roboczej
 - Drukarka
 - * Podłączenie drukarki
- Software
 - e-mail
 - * Utworzenie konta
 - Aplikacja
 - * Instalacja oprogramowania
 - * Usunięcie oprogramowania
 - * Aktualizacja oprogramowania
 - * Nadanie uprawnień.
 - * Odebranie uprawnień.

Rozdział 4

Wymagania

W tym rozdziale opisano wymagania dla realizowanego fragmentu systemu Service Desk. Wymagania dotyczą wybranych procesów przedstawionych w sekcji 2.7.

4.1 Wymagania funkcjonalne

4.1.1 Zarządzanie incydentami

- IM.1 – System Service Desk powinien umożliwiać zgłoszanie nowych incydentów klientom usługi.
- IM.2 – Pracownik pierwszej linii powinien mieć możliwość przypisania incydentu do siebie.
- IM.3 – System powinien umożliwiać sprawdzenie usług, z których korzysta klient.
- IM.4 – Przy rozwiązywaniu incydentu pracownik pierwszej linii powinien mieć możliwość wyszukiwania gotowych rozwiązań w bazie znanych błędów.
- IM.5 – System powinien umożliwiać dodawanie komentarzy do incydentu wszystkim zainteresowanym.
- IM.6 – System powinien przechowywać pełną historię incydentu.
- IM.7 – System powinien umożliwiać skonfigurowanie reguł eskalacji z uwzględnieniem umów SLA.
- IM.8 – System powinien umożliwiać wyszukiwanie incydentów.

- IM.9 – Klient usługi powinien mieć możliwość przeglądania wszystkich incydentów, które zgłosił.
- IM.10 – System powinien powiadamiać klienta o zmianie statusu incydentu.
- IM.11 – System powinien powiadamiać serwisanta oraz menedżera grupy o nowo przypisanych incydentach.

4.1.2 Zarządzanie problemami

- PM.1 – System powinien umożliwiać zgłaszanie nowych problemów.
- PM.2 – System powinien umożliwiać powiązanie incydentów z problemem.
- PM.3 – System powinien umożliwiać dodawanie rozwiązań tymczasowych oraz stałych dla problemów.
- PM.4 – System powinien umożliwiać dodawanie znanych błędów do KEDB.
- PM.5 – System powinien umożliwiać wyszukiwanie problemów.

4.1.3 Zarządzanie zdarzeniami

- EM.1 – System powinien udostępniać interfejs do dodawania zdarzeń innym systemom.
- EM.2 – System powinien umożliwiać automatyczne wygenerowanie incydentu dla wybranych zdarzeń.
- EM.3 – System powinien umożliwiać wyszukiwanie zdarzeń.

4.1.4 Zarządzanie zleceniami

- RF.1 – System powinien umożliwiać konfigurację kategorii standardowych zmian.
- RF.2 – System powinien umożliwiać zgłaszanie standardowych zmian przez klientów usługi.

4.2 Wymagania niefunkcjonalne

4.2.1 NF.WOL – Wolumetria

- NF.WOL.1 – System powinien obsłużyć co najmniej 1 000 000 incydentów.
- NF.WOL.2 – System powinien obsłużyć co najmniej 10 000 pracowników oraz klientów.

- NF.WOL.3 – System powinien zakładać przyszły wzrost obciążenia i być dostosowany do przeskalowania.

4.2.2 NF.SPR – Sprawność i efektywność

- NF.SPR.1 – Czas odpowiedzi aplikacji, w 99% przypadków nie powinien przekroczyć 1 sekundy.
- NF.SPR.2 – System powinien umożliwiać jednoczesną pracę co najmniej 10 pracowników.

4.2.3 NF.SEC – Bezpieczeństwo (security)

- NF.SEC.1 – Dostęp do systemu może uzyskać tylko uwierzytelniony użytkownik.
- NF.SEC.2 – Użytkownik posiada dostęp tylko do danych i funkcji systemu niezbędnych do wykonywanej przez niego pracy.
- NF.SEC.3 – System musi być odporny na awarie pojedynczych nośników dyskowych.

4.2.4 NF.SAF – Bezpieczeństwo (safety)

- NF.SAF.1 – Historia kopii zapasowych powinna umożliwić odtworzenie danych sprzed ostatnich 10 lat.

4.2.5 NF.NZW – Niezawodność i dostępność

- NF.NZW.1 – Żadna akcja użytkownika systemu nie może powodować niedostępności systemu.
- NF.NZW.2 – Żadna odpowiedzialna akcja administratora systemu nie powinna powodować niedostępności systemu.
- NF.NZW.3 – Dostępność systemu musi wynosić co najmniej 99.99% w ciągu roku w godzinach pracy.
- NF.NZW.4 – Średni czas do odtworzenia systemu po awarii (MTTR) powinien być mniejszy niż 4h w godzinach pracy. RTO¹ jedynie w przypadku katastrofy ośrodka przetwarzania może wynieść jeden dzień roboczy.
- NF.NZW.5 – Utrata zatwierdzonych danych w przypadku awarii powinna wynosić 0 (RPO - Recovery Point Objective), a w przypadku katastrofy ośrodka przetwarzania nie więcej niż jeden dzień roboczy.

¹RTO – Recovery Time Objective

- NF.NZW.6 – System musi być odporny na niezależne awarie nośników dyskowych (MTTF co najmniej 1 000 000 lat).

4.2.6 NF.ERG – Ergonomia

- NF.ERG.1 – Wszystkie interfejsy użytkownika powinny być utrzymane w jednolitej konwencji graficznej.
- NF.ERG.2 – System powinien być poprawnie użytkowany po najwyżej 24 godzinach szkolenia. Liczba błędów wykonanych przez użytkownika nie powinna przekraczać 2 dziennie.

4.2.7 NF.STA – Standardy używane przez system

- NF.STA.1 – Kodowanie znaków w systemie musi być w standardzie UTF-8.
- NF.STA.2 – System musi umożliwiać zapisywanie generowanych raportów do formatu PDF (ISO 32000-1:2008).

4.2.8 NF.PRZ – Przenośność

- NF.PRZ.1 – System powinien wspierać przeglądarki internetowe w podanych wersjach: Firefox 3.x oraz Google Chrome 3.x.
- NF.PRZ.2 – Przeglądarka nie powinna wymagać żadnej konfiguracji poza wyłączeniem obsługi technologii Cookies i JavaScript.

4.2.9 Konkluzja

Wymagania niezawodności i dostępności nie mogą być zapewnione przez sam projekt, ale wymagają odpowiednich działań na etapie eksploatacji, przykładowo: codzienne składowanie kopii zapasowej i archiwów na zewnątrz ośrodka przetwarzania. Na etapie projektu można jedynie wybrać właściwe narzędzia oraz komponenty systemu, co zostało dokonane w ramach pracy. Opis implementacji systemu, zawierający omówienie wybranych narzędzi oraz technologii, przedstawiono w rozdziale 6. Przygotowanie procedur eksploatacyjnych wykracza poza zakres niniejszej pracy.

Podjęte decyzje projektowe, wybrane narzędzia, komponenty systemu, odpowiedni sprzęt oraz właściwe procedury na etapie eksploatacji zapewnią realizację wymagań niefunkcjonalnych.

4.3 Elementy realizujące wymagania

Niniejsza sekcja przedstawia elementy realizujące wymagania funkcjonalne opisane w sekcji 4.1.

4.3.1 Zarządzanie zdarzeniami

- EM.WS – Web Service umożliwiający dodawanie listy zdarzeń (ang. Event). Każde ze zdarzeń może wygenerować incydent, wtedy zdarzenie zostanie automatycznie powiązane z tym incydentem.
- EM.CARD – Karta ze szczegółami zdarzenia. Dla każdego zdarzenia prezentowane są informacje o identyfikatorze, dacie zdarzenia, dacie dodania do systemu, randze, kategorii oraz temacie. Dodatkowo każde zdarzenia posiada listę parametrów w postaci: klucz, wartość. Dla zdarzeń, które wygenerowały incydent jest odnośnik do karty incydentu.
- EM.SEARCH – Wyszukiwanie zdarzeń po identyfikatorze, zakresie daty zgłoszenia, zakresie daty wystąpienia, temacie, randze i kategorii.
- EM.REPORT – Raportowanie dotyczące zdarzeń
 - EM.REPORT.CAT – Raport zdarzeń wg kategorii.
 - EM.REPORT.SIG – Raport zdarzeń wg rangi zdarzenia.
 - EM.REPORT.INC – Raport zdarzeń, które wygenerowały incydent.

4.3.2 Zarządzanie incydentami

- IM.ADD – Dodawanie incydentów. Należy podać temat incydentu oraz opis. Przy braku podania dowolnego z pól system zgłosi wyjątek.
 - IM.ADD.FOR – Dodanie incydentu dla dowolnego pracownika. Funkcjonalność dostępna tylko dla pracowników pierwszej linii wsparcia. W takim wypadku należy wybrać źródło zgłoszenia: e-mail lub telefon.
- IM.CARD – Karta ze szczegółami incydentu. Dla każdego incydentu prezentowane są informacje o identyfikatorze, przypisanej usłudze, priorytecie, statusie, osobie, która zgłosiła incydent, dacie zgłoszenia, terminie rozwiązania, dacie rozwiązania, dacie zamknięcia, temacie oraz opisie. Termin rozwiązania jest wyliczany na podstawie daty dodania oraz umowy SLA². Data rozwiązania jest ustawiona automatycznie, gdy incydent zmieni status na rozwiązany (ang. Resolved), analogicznie jest z datą zamknięcia. Dodatkowo prezentowane są dane kontaktowe osoby zgłaszającej incydent: telefon, telefon komórkowy i adres e-mail.
 - IM.CARD.ADD.PM – Dodanie problemu z poziomu karty incydentu. Dodany problem automatycznie zostanie powiązany z incydentem.

²SLA – Service Level Agreement

- IM.CARD.COMMENT – Dodanie komentarza na karcie incydentu.
- IM.CARD.ASSIGN – Przypisanie incydentu do aktualnie zalogowanego użytkownika.
- IM.CARD.SERVICES – Prezentowane są usługi, z których korzysta klient.
- IM.SEARCH – Wyszukiwanie incydentów po identyfikatorze, zakresie daty zgłoszenia, fragmencie tematu, osobie zgłaszającej, statusie, priorytecie, wpływie, kategorii, serwisancie oraz grupie wsparcia.
- IM.MYOPEN – Widok z listą otwartych incydentów dla aktualnie zalogowanego użytkownika.
- IM.MYCLOS – Widok z listą zamkniętych incydentów dla aktualnie zalogowanego użytkownika.
- IM.NOTASSIGN – Widok z listą incydentów nie przypisanych do żadnego serwisanta.
- IM.MYASSIGN – Widok z listą incydentów przypisanych do aktualnie zalogowanego pracownika.
- IM.REPORT – Raportowanie dotyczące incydentów
 - IM.REPORT.CAT – Raport incydentów wg kategorii
 - IM.REPORT.SRC – Raport incydentów wg źródła
 - IM.REPORT.PRI – Raport incydentów wg priorytetu
 - IM.REPORT.SG – Raport incydentów wg grupy wsparcia oraz serwanta
- IM.SG.DETAILS – Widok ze szczegółowymi danymi na temat grupy wsparcia. Prezentowane są następujące dane: nazwa grupy, manager grupy, lista członków grupy.
- IM.HISTORY – Prezentowanie historii incydentu w postaci listy akcji określonego typu.
- IM.SG.MAN – Stworzenie formularza master-detail do zarządzania grupami wsparcia oraz listą pracowników w grupie.

4.3.3 Zarządzanie problemami

- PM.ADD – Dodawanie problemów. Należy podać temat problemu oraz opis. Przy braku podania dowolnego z pól system zgłosi wyjątek.

- PM.CARD – Karta ze szczegółami problemu. Dla każdego problemu prezentowane są informacje o identyfikatorze, dacie zgłoszenia, dacie zamknięcia, serwisancie, grupie wsparcia, osobie zgłaszającej, temacie oraz opisie.
 - PM.CARD.COMMENT – Dodanie komentarza na karcie problemu.
- PM.SEARCH – Wyszukiwanie problemów po identyfikatorze, zakresie daty zgłoszenia, fragmencie tematu, osobie zgłaszającej, statusie, priorytecie, wpływie, kategorii, serwisancie i grupie wsparcia.
- PM.KEDB – Zarządzanie znymi błędami
 - PM.KEDB.SEARCH – Wyszukiwanie znanych błędów po identyfikatorze, temacie, kategorii, symptomach, głównej przyczynie oraz rozwiązaniu tymczasowym.
 - PM.KEDB.KE – Zamiana problemu w znany błąd (ang. known error). Funkcjonalność dostępna dla menedżera problemów.
- PM.SOL – Dodanie rozwiązania oraz rozwiązania tymczasowego dla problemu.
- PM.ASSOC.I – Powiązanie incydentu z problemem.

4.3.4 Zarządzanie zleceniami

- RF.ADD – Dodawanie zleceń. Wybranie kategorii zlecenia. Każda kategoria zawiera listę parametrów, które należy podać. Przykładowo: zgłoszenie instalacji nowego oprogramowania może zawierać nazwę aplikacji oraz wersję aplikacji.
- RF.CARD – Karta szczegółów zlecenia. Dla każdego zlecenia prezentowane są informacje o identyfikatorze zgłoszenia, osobie, która dodała zgłoszenie, kategorii, statusie, terminie rozwiązania, dacie zgłoszenia, dacie zamknięcia, serwisancie oraz grupie. Dodatkowo istnieje lista parametrów zależnych od kategorii zlecenia.
 - RF.CARD.COMMENT – Dodanie komentarza na karcie zlecenia.
- RF.CONF – Stworzenie formularza master-detail do zarządzania kategoriemi zleceń oraz listą parametrów zlecenia. Użytkownik z rolą menedżera zleceń ma możliwość dodawania nowych kategorii zleceń i ustalania parametrów tych kategorii.

4.3.5 Administracja

- ADMIN.SE – Wyszukiwanie pracowników po identyfikatorze, loginie, imieniu oraz nazwisku.

4.3.6 Zarządzanie pracownikami

- EMP.DETAILS – Widok ze szczegółowymi danymi na temat pracownika lub klienta: imię, nazwisko, telefon komórkowy, telefon, e-mail, department, stanowisko i menedżer pracownika. Dodatkowo prezentowane są usługi, z których korzysta pracownik lub klient.
- EMP.ROLES – Administrator systemu ma możliwość zarządzania rolami pracownika.

4.3.7 Zarządzanie sygnałami

- SIG.TRG – Wygenerowanie sygnału określonego typu w wyniku zmiany stanu zdarzenia. Przykładowo wygenerowanie sygnału po zmianie statusu incydentu na rozwiązany. Sytuacje, w których generowany jest sygnał są z góry ustalone w aplikacji.
- SIG.EMAIL – Zadanie (job) wysyłające wiadomości e-mail dla dodanych sygnałów. Treść e-maila jest tworzona na podstawie szablonu typu sygnału, analogicznie z tematem wiadomości e-mail. Wiadomość e-mail jest wysyłana do osoby, dla której został wygenerowany sygnał.
- SIG.TYPE – Zarządzanie typami sygnałów. Dla każdego typu powinien zostać określony szablon wiadomości, wypełniany za pomocą atrybutów obiektu. Analogicznie jest z tematem wiadomości. Każdy typ sygnału powinien być określonej rangi: informacja, ostrzeżenie lub błąd.
 - SIG.TYPE.F – Stworzenie formularza do zarządzania typami sygnałów.
 - SIG.TYPE.F.MSG – Podanie szablonu wiadomości.
 - SIG.TYPE.F.SUB – Podanie szablonu tematu wiadomości.
- SIG.HOME – Prezentacja sygnałów użytkownika na stronie głównej.
- SIG.GEN – Zarządzanie generatorami sygnałów. Każdy generator jest uruchamiany co pewien kwant czasu. Generator dotyczy konkretnego obiektu: incydentu, problemu lub zlecenia. Sprawdza, czy istnieją obiekty spełniające podane kryterium. Dla zadanego kryterium generuje sygnał określonego typu dla określonego pracownika, który ustalany jest na zasadzie połączenia z tym obiektem. Przykładowo pracownik może być osobą, która zgłosiła incydent, menedżerem tego pracownika, serwisantem, menedżerem grupy, menedżerem serwisanta itd.
 - SIG.GEN.F – Stworzenie formularza do zarządzania generatorami.
 - SIG.GEN.F.ADD – Możliwość dodawania nowych generatorów.

- SIG.GEN.F.DEL – Możliwość usuwania generatorów.
- SIG.GEN.F.EDIT – Możliwość edycji generatorów.
- SIG.GEN.F.TIME – Ustawienie częstotliwości generatora.
- SIG.GEN.F.ON – Włączenie/wyłączenie generatora.
- SIG.GEN.F.EMP – Wybranie kategorii pracownika z ustalonej listy: serwisant, zgłaszający, menedżer grupy, menedżer grupy nadzędnej, menedżer serwisanta lub menedżer zgłaszającego.
- SIG.GEN.F.CRIT – Podanie kryterium w polu tekstowym.
- SIG.GEN.F.PREVIEW – Podgląd obiektów aktualnie spełniających kryterium zadane przez generator.

Rozdział 5

Projekt

W tym rozdziale przedstawiono projekt bazy danych utworzony na podstawie wymagań zebranych w rozdziale 4. Został również zaprezentowany projekt aplikacji obejmujący omówienie warstw aplikacji. Na podstawie przykładowego diagramu sekwencji zaprezentowano współpracę pomiędzy poszczególnymi warstwami aplikacji. Ostatecznie omówiono zaproponowane rozwiązanie dla eskalacji: sygnały.

5.1 Baza danych

Opis bazy danych został podzielony ze względu na wybrane procesy. Dla każdego procesu zostały przedstawione wykorzystywane tabele w tym procesie.

5.1.1 Zarządzanie incydentami

INCIDENTS

Tabela przechowująca informacje o incydentach.

CATEGORIES_IM

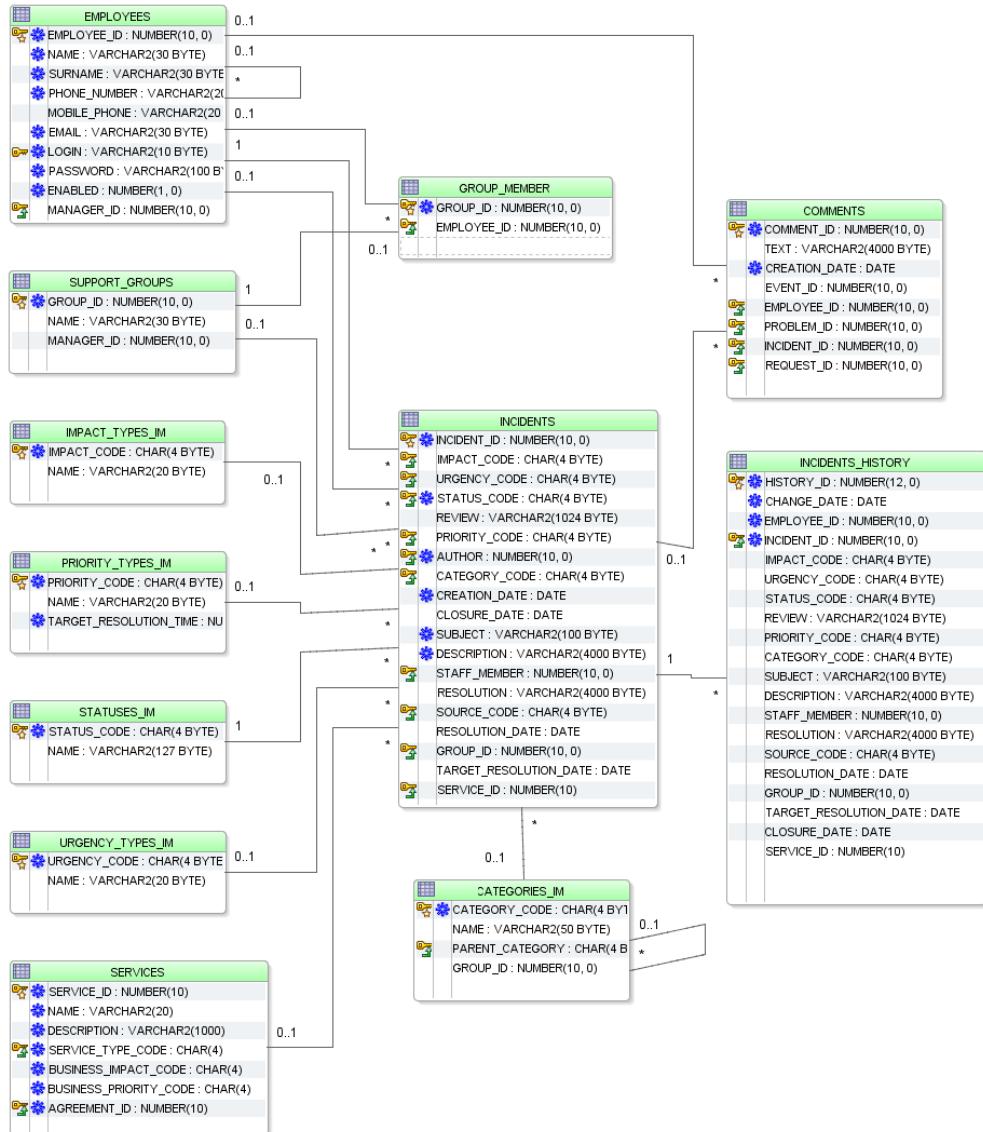
Słownik z kategoriami incydentów.

URGENCY_TYPES_IM

Słownik z typami pilności incydentów.

STATUSES_IM

Słownik z możliwymi statusami incydentów.



Rysunek 5.1: Diagram tabel dla zarządzania incydentami

PRIORITY_TYPES_IM

Słownik z typami priorytetów incydentów.

IMPACT_TYPES_IM

Słownik z typami wpływów incydentów.

COMMENTS

Tabela z komentarzami dotyczącymi danego incydentu. Każdy komentarz zawiera datę dodania oraz autora.

EMPLOYEES

Tabela z pracownikami. Każdy incydent powiązany jest z pracownikami na dwa sposoby:

- pracownik, który zgłosił incydent,
- serwisant przypisany do incydentu.

SUPPORT_GROUPS

Tabela z grupami wsparcia. Każdy incydent jest przypisany do jednej grupy wsparcia.

SERVICES

Tabela zawierająca dane o usługach. Każdy incydent jest przypisany do jednej usługi.

INCIDENT_HISTORY

Tabela zawierająca pełną historię zmian dla każdego incydentu. Zawiera datę dokonania zmiany, użytkownika, który dokonał zmiany oraz stan incydentu z tej chwili w czasie.

5.1.2 Zarządzanie problemami

PROBLEMS

Tabela przechowująca informacje o problemach.

CATEGORIES_PM

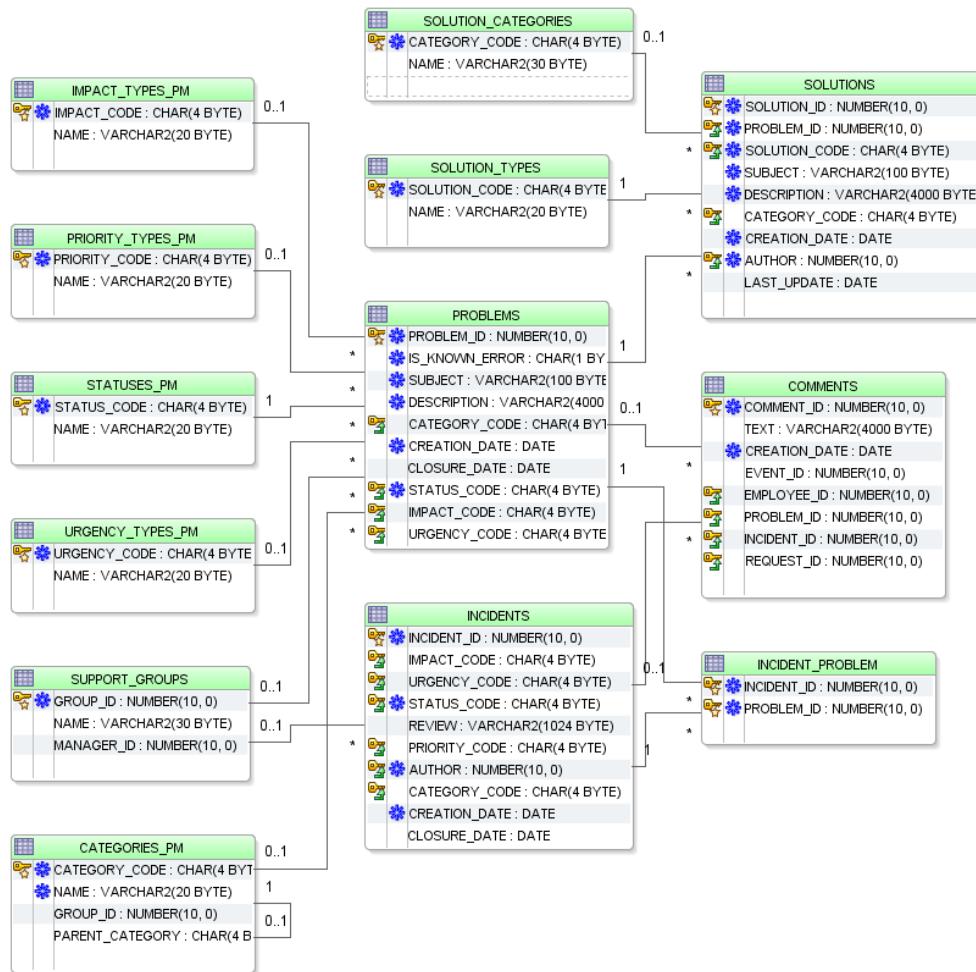
Słownik z kategoriami problemów.

URGENCY_TYPES_PM

Słownik z typami pilności problemów.

STATUSES_PM

Słownik z możliwymi statusami problemów.



Rysunek 5.2: Diagram tabel dla zarządzania problemami

PRIORITY_TYPES_PM

Słownik z typami priorytetów problemów.

IMPACT_TYPES_PM

Słownik z typami wpływów problemów.

COMMENTS

Tabela z komentarzami dotyczącymi danego problemu. Każdy komentarz zawiera datę dodania oraz autora.

EMPLOYEES

Tabela z pracownikami. Każdy problem powiązany jest z pracownikami na dwa sposoby:

- pracownik, który zgłosił problem,
- serwisant przypisany do problemu.

SUPPORT_GROUPS

Tabela z grupami wsparcia. Każdy problem jest przypisany do jednej grupy wsparcia.

INCIDENT_PROBLEM

Tabela asocjacyjna reprezentująca powiązania pomiędzy problemami oraz incydentami. Każdy problem jest powiązany z listą incydentów.

SOLUTIONS

Tabela zawierająca rozwiązania problemów. Są to zarówno rozwiązania tymczasowe¹ oraz rozwiązania stałe.

SOLUTION_TYPES

Słownik z typami rozwiązań problemów. Standardowo powinny to być dwa rodzaje rozwiązań: tymczasowe, stałe. Aczkolwiek system umożliwia dodanie nowych typów rozwiązań.

5.1.3 Zarządzanie zdarzeniami

EVENTS

Tabela przechowująca informacje o zdarzeniach.

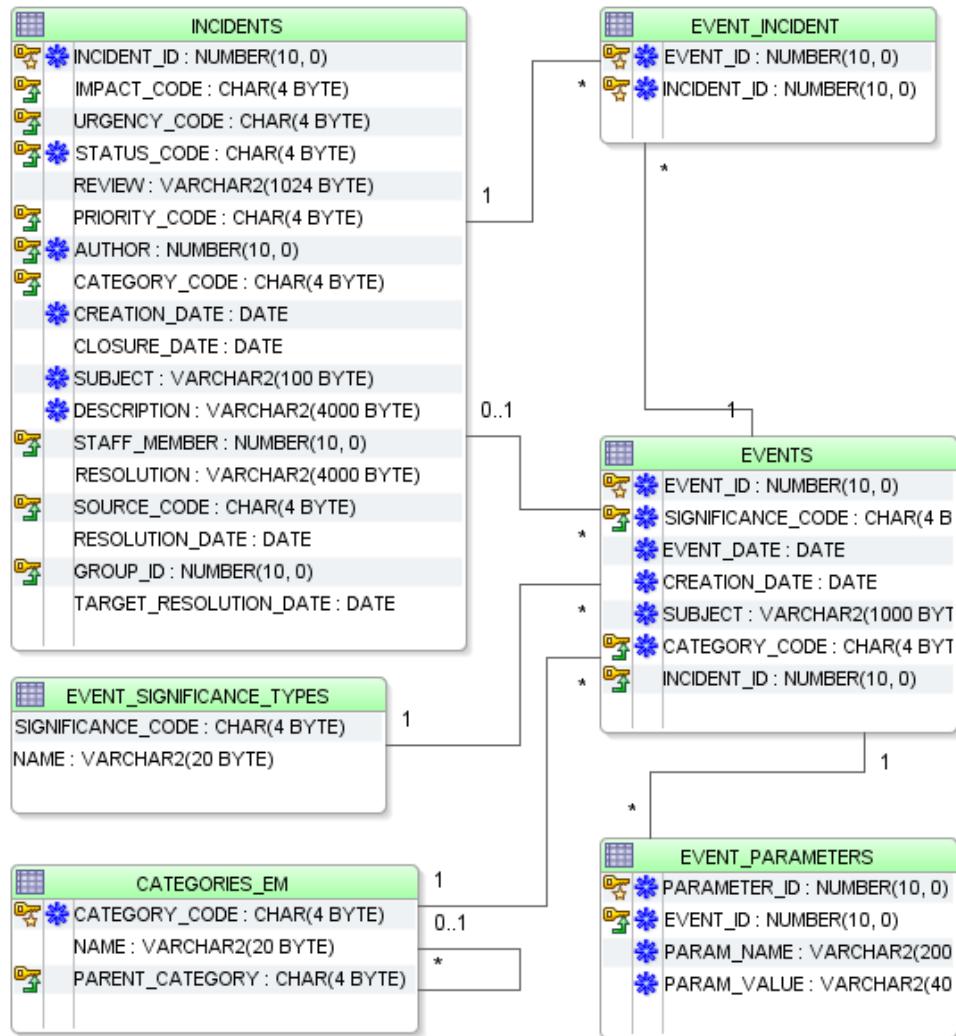
CATEGORIES_EM

Słownik z kategoriami zdarzeń.

EVENT_SIGNIFICANCE_TYPES

Słownik z możliwymi rangami zdarzeń. Przykładowe rangi: informacyjna, ostrzeżenie, błąd.

¹Rozwiązanie tymczasowe - ang. Workaround



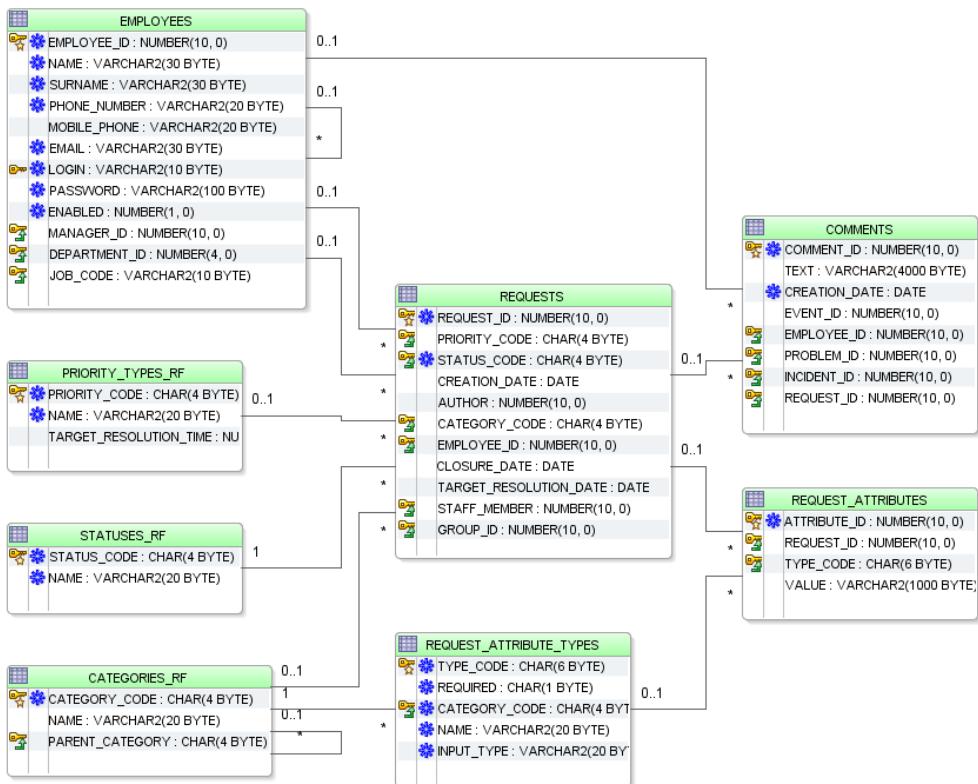
Rysunek 5.3: Diagram tabel dla zarządzania zdarzeniami

EVENT_PARAMETERS

Tabela z listą parametrów danego zdarzenia. Każdy parametr jest postaci klucz, wartość.

EVENT INCIDENT

Tabela asocjacyjna zawierająca powiązania pomiędzy zdarzeniami oraz incydentami.



Rysunek 5.4: Diagram tabel dla zarządzania zleceniami

5.1.4 Zarządzanie zleceniami

REQUESTS

Tabela przechowująca informacje o zleciennach.

CATEGORIES_RF

Słownik z kategoriami zleceń. Każda kategoria zawiera listę atrybutów w tabeli REQUEST_ATTRIBUTE_TYPES.

STATUSES_RF

Słownik ze statusami zleceń.

PRIORITY_TYPES_RF

Słownik z priorytetami zleceń.

COMMENTS

Tabela z komentarzami dotyczącymi danego zlecenia. Każdy komentarz zawiera datę dodania oraz autora.

EMPLOYEES

Tabela z pracownikami. Każde zlecenie powiązany jest z pracownikami na dwa sposoby:

- pracownik, który zgłosił zlecenie,
- serwisant przypisany do zlecenia.

SUPPORT_GROUPS

Tabela z grupami wsparcia. Każde zlecenie jest przypisane do jednej grupy wsparcia.

REQUEST_ATTRIBUTE_TYPES

Tabela z typami atrybutów poszczególnych kategorii zleceń.

REQUEST_ATTRIBUTES

Tabela z wartościami atrybutów dla zleceń. Każdy atrybut jest określonego typu, zawartego w tabeli REQUEST_ATTRIBUTE_TYPES.

5.1.5 Zarządzanie sygnałami

SIGNALS

Tabela przechowująca informacje o sygnałach.

SIGNAL_TYPES

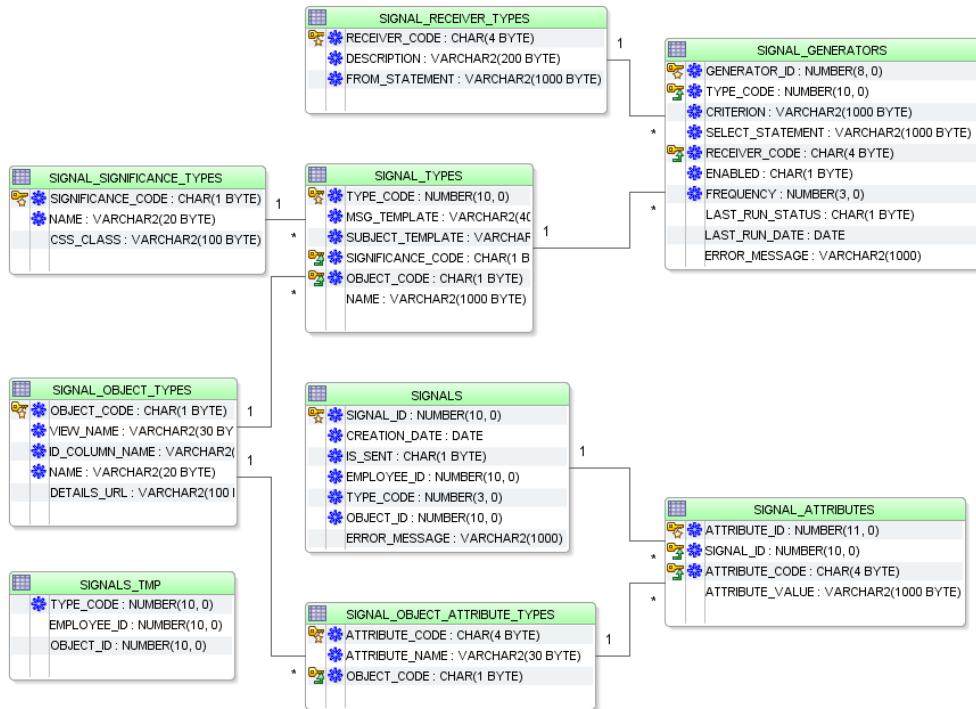
Tabela z typami sygnałów. Każdy typ sygnału jest przypisany do typu obiektu.

SIGNAL_ATTRIBUTES

Tabela z listą wartości atrybutów dla konkretnego sygnału. Każdy atrybut jest typu zawartego w tabeli: SIGNAL_OBJECT_ATTRIBUTE_TYPES.

SIGNAL_GENERATORS

Tabela z generatorami sygnałów. Każdy generator tworzy sygnały określonego typu. Tabela zawiera kryterium, które muszą spełniać obiekty oraz powiązania z kategorią odbiorcy z tabeli SIGNAL_RECEIVER_TYPE.



Rysunek 5.5: Diagram tabel dla sygnałów

SIGNAL_RECEIVER_TYPES

Tabela z kategoriami odbiorców sygnałów. Odbiorca jest określany dynamicznie na podstawie powiązania z danym obiektem: incydentem, problemem, zleceniem. Przykładowe typy odbiorców:

- pracownik, który zgłosił incydent,
- serwisant przypisany do incydentu,
- menedżer serwisanta,
- menedżer grupy wsparcia przypisanej do incydentu.

SIGNAL_OBJECT_TYPES

Tabela z typami obiektów, dla których może zostać wygenerowany sygnał. Przykładowe typy: incydent, problem, zlecenie. Każdy typ zawiera listę atrybutów w postaci klucz, wartość. Atrybutu wykorzystywane są przy szablonie treści wiadomości oraz szablonie tematu wiadomości. Każdy typ związany jest z perspektywą, zawierającą szczegółowe dane o atrybutach obiektu.

SIGNALS_TMP

Tabela tymczasowa zawierająca sygnały, które należy dodać.

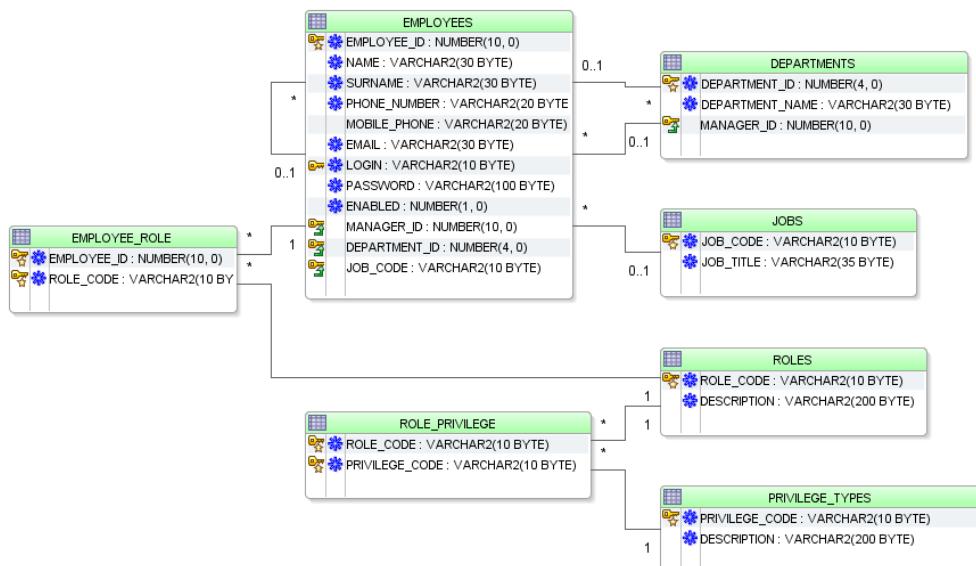
SIGNAL_OBJECT_ATTRIBUTE_TYPES

Tabela z listą atrybutów dla typu obiektu. Przykładowe atrybuty: data zgłoszenia incydentu, serwisant przypisany do incydentu lub kategoria.

SIGNAL_SIGNIFICANCE_TYPES

Słownik z rangami sygnałów. Przykładowe rangi to: informacyjna, ostrzeżenie, błąd.

5.1.6 Zarządzanie pracownikami



Rysunek 5.6: Diagram tabel dla zarządzania pracownikami

EMPLOYEES

Tabela przechowująca informacje o pracownikach.

JOBS

Słownik z możliwymi stanowiskami pracowników.

DEPARTMENTS

Tabela z działami, do których przypisani są pracownicy.

ROLES

Słownik z możliwymi rolami aplikacyjnymi. Przykładowe role: pracownik pierwszej linii, menedżer incydentów, menedżer problemów oraz administrator.

EMPLOYEE_ROLE

Tabela zawierająca role przypisane dla danego pracownika.

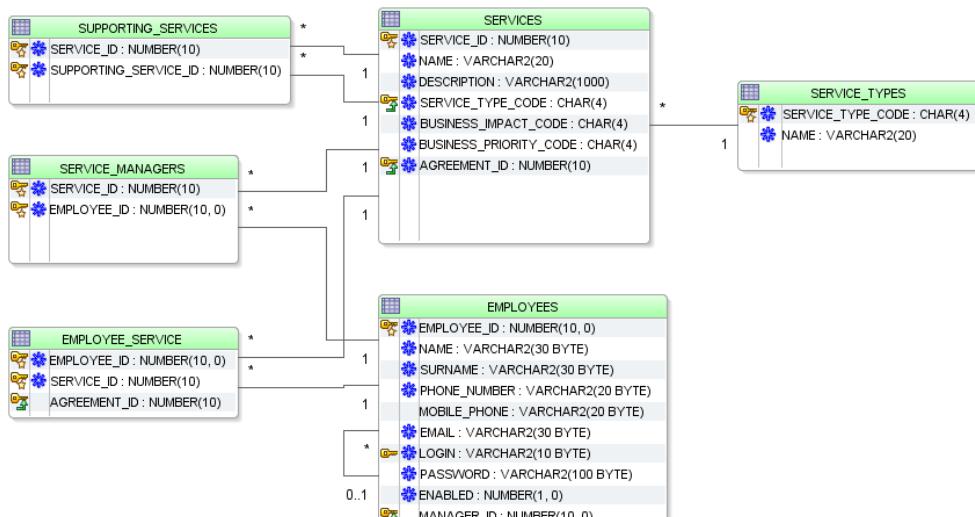
PRIVILEGE_TYPES

Tabela zawierająca uprawnienia aplikacyjne. Przykładowe uprawnienia: dodanie incydentu, dodanie problemu.

ROLE_PRIVILEGES

Tabela zawierająca uprawnienia przypisane do danej roli aplikacyjnej.

5.1.7 Zarządzanie katalogiem usług



Rysunek 5.7: Diagram tabel dla zarządzania katalogiem usług

SERVICES

Tabela zawierająca dane o usługach.

SUPPORTING_SERVICES

Tabela zawierająca dane o zależnościach pomiędzy usługami. Każda usługa biznesowa jest oparta na grupie usług technicznych.

SERVICE_MANAGERS

Tabela zawierająca listę menedżerów danej usługi.

SERVICE_TYPES

Słownik z typami usług. Przykładowe typy: usługa biznesowa, usługa techniczna.

EMPLOYEE_SERVICE

Tabela zawierająca usługi przypisane do danego pracownika.

5.1.8 Zarządzanie poziomem usług



Rysunek 5.8: Diagram tabel dla zarządzania poziomem usług

AGREEMENTS

Tabela zawierająca umowy: SLA, OLA, UC.

EMPLOYEE_SERVICE

Tabela zawierająca każdą indywidualną umowę SLA dla danego klienta.

5.2 Aplikacja

5.2.1 Warstwy aplikacji

Aplikacja ma budowę warstwową. Każda z warstw jest zależna od sąsiednich oraz niezależna od pozostałych. Umożliwia to łatwiejszą wymianę dowolnej z warstw, bez dokonywania ingerencji w resztę.

Warstwa prezentacji

Prezentacja została wykonana w technologiach HTML, CSS oraz JavaScript. Raporty są prezentowane w formacie PDF za pomocą biblioteki JasperReports.

Kontrolery stron

Każde żądanie jest obsługiwane przez przypisany kontroler strony. Kontroler przy pomocy klas z warstwy usługowej wykonuje żądanie oraz przy pomocy klas z warstwy validacji sprawdza poprawność parametrów.

Warstwa usługowa

W tej warstwie znajdują się klasy usługowe, wykonujące żądania na polecenie kontrolerów. Wykorzystywane są klasy z warstwy dostępu do danych w celu pobierania oraz zapisywania obiektów.

Warstwa dostępu do danych

Jest to warstwa odpowiedzialna za pobieranie oraz zapisywanie obiektów w bazie danych.

Warstwa domenowa

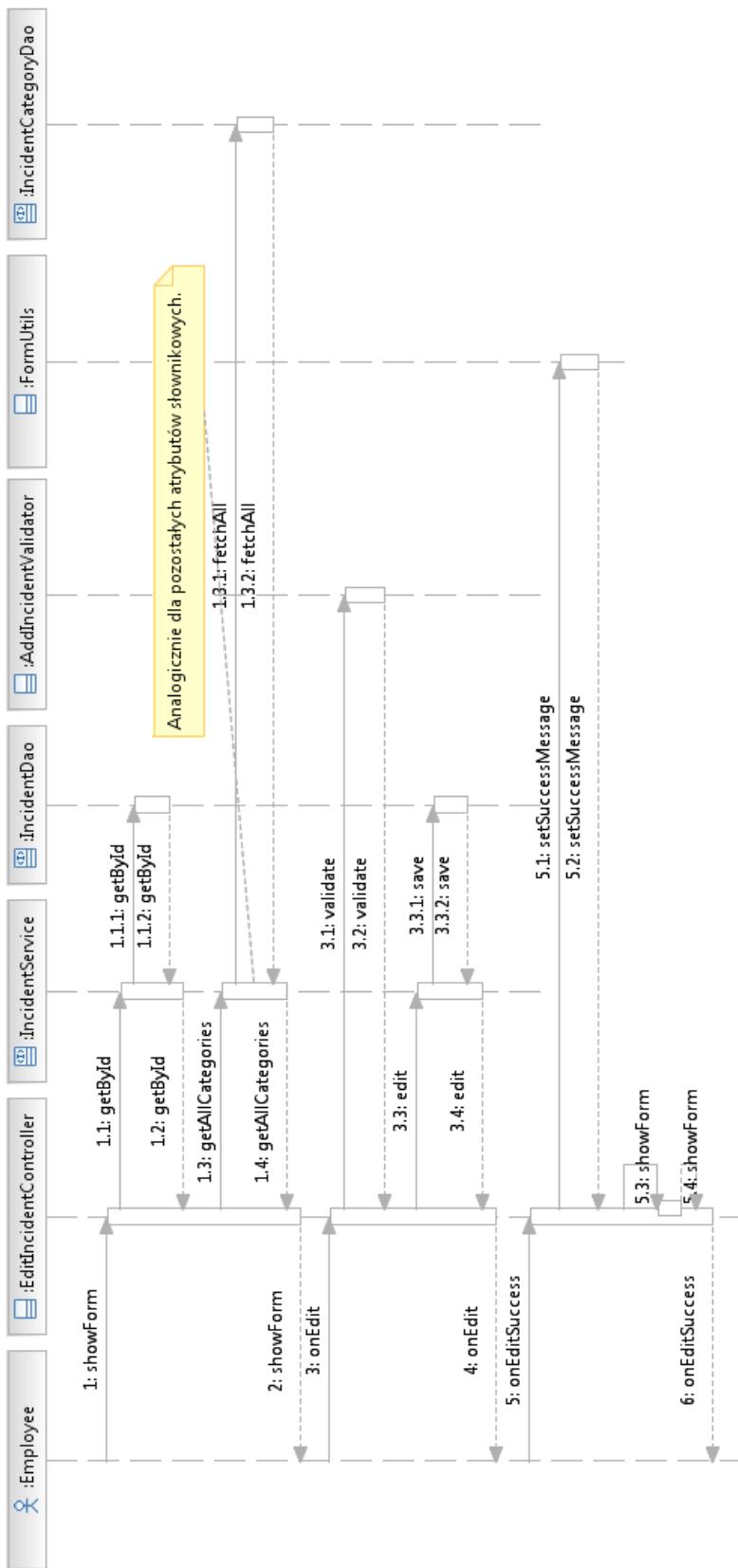
W tej warstwie są klasy reprezentujące model danych.

Warstwa validacji

Jest to warstwa zajmująca się sprawdzaniem poprawności parametrów żądania.

5.2.2 Przykładowy diagram sekwencji

Rysunek 5.9 przedstawia przykładowy diagram sekwencji dotyczący edycji incydentu. Na rysunku widoczne są relacje pomiędzy klasami z różnych warstw



Rysunek 5.9: Diagram sekwencji dotyczący edycji incydentu

aplikacji. Klasa `EditIncidentController` jest to kontroler stron. Obsługuje żądania od użytkownika aplikacji. Korzysta z klasy `IncidentService`, która zawiera podstawowe usługi wyszukiwania, zapisywania. `EditIncidentController` korzysta również z klasy `AddIncidentValidator` do przeprowadzenia walidacji. `IncidentService` jest klasą z warstwy usługowej, dostarcza usługi dla klasy `EditIncidentController` oraz korzysta z klas dostępu do danych: `IncidentDao` oraz `IncidentCategoryDao`, które są odpowiedzialne za zapisywanie i pobieranie klas domenowych z bazy danych.

W pierwszym kroku użytkownik wyświetla kartę incydentu. W aplikacji jest to realizowane jako wywołanie metody `showForm()` klasy `EditIncidentController`. Klasa `EditIncidentController` musi pobrać dane potrzebne do wyświetlenia karty incydentu, korzystając z usług udostępnionych przez klasę `IncidentService`. Pierwsza metoda `getById()` pobiera incydent po identyfikatorze. Klasa `IncidentService` nie przeprowadza operacji w bazie danych, natomiast korzysta z klas z warstwy dostępu do danych. W tym przypadku wywoła metodę `getById()` klasy `IncidentDao`. Należy również pobrać wartości wszystkich słowników, w tym przypadku klasa `IncidentService` również udostępnia szereg metod odpowiedzialnych za wykonanie tej czynności. Diagram przedstawia pobranie wszystkich możliwych kategorii incydentów. `IncidentService` zawiera metodę `getAllCategories()`, która wykona tę czynność. Tutaj podobnie wykorzystywana jest klasa dostępu do danych `IncidentCategoryDao`, zawierająca metodę `fetchAll()`.

W drugim kroku użytkownik dokonuje zapisania formularza, co powoduje wywołanie metody `onEdit()` klasy `EditIncidentController`. Metoda ta na wstępie dokonuje walidacji, przy pomocy klasy `AddIncidentValidator`, a następnie zapisania incydentu przy pomocy metody `edit()` klasy `IncidentService`, co spowoduje wywołanie metody `save()` klasy `IncidentDao`.

W trzecim kroku zostaje wyświetlony ten sam formularz z komunikatem o sukcesie.

Przykład ten jest reprezentatywny dla całej aplikacji. Wszystkie żądania w aplikacji działają wg przedstawionego schematu.

5.2.3 Aspekty aplikacji

Przy implementacji aplikacji skorzystano z programowania aspektowego². Zostały utworzone aspekty odpowiedzialne za obszary: bezpieczeństwa oraz spójności transakcyjnej.

Programowanie aspektowe, wykorzystując proces wplatania (ang. weaving), pozwala na zaaplikowanie nowego kodu do kodu już istniejącego. W typowym podejściu należałoby własnoręcznie dodać nowy kod do istniejącego kodu, na-

²AOP - Aspect Oriented Programming

tomiast programowanie aspektowe (w procesie wplatania) wykonuje tę czynność automatycznie [9].

Aspekt bezpieczeństwa

Aspekt bezpieczeństwa umożliwia zarządzanie poziomem uprawnień w aplikacji bez ingerencji w logikę biznesową. Przy pomocy adnotacji *@Secured* można oznaczyć wybraną metodę w celu wskazania, które role aplikacyjne mają prawo do jej wykonania. Istnieje również możliwość oznaczenia wybranych metod w pliku konfiguracyjnym XML [13].

Aspekt spójności transakcyjnej

Aspekt spójności transakcyjnej zapewnia spójność transakcyjną poprzez oznaczenie wybranej metody adnotacją *@Transactional*. Gwarantuje to wykonanie treści tej metody w ramach jednej transakcji bazodanowej. Twórca aplikacji oznaczył tą adnotacją wszystkie metody z warstwy usługowej [13].

5.3 Sygnały

Wymagania narzucają powiadomianie odpowiednich osób o zmianie stanu zdarzenia: incydentu, problemu oraz zlecenia. Przykładem jest informacja dla klienta o rozwiązaniu incydentu lub informacja dla serwisanta o nowo przypisany incydencie. Pokrewnym wymaganiem jest eskalacja. Po określonym upływie czasu należy poinformować odpowiednią osobę.

Oba typy powiadomień mają wiele cech wspólnych, aczkolwiek różnią się rodzajem wyzwalacza. Pierwsze z nich są wyzwalane przez zmianę stanu, a drugie poprzez czas. Autor niniejszej pracy nazwał oba rodzaje powiadomień sygnałami.

Sygnal jest to powiadomienie użytkownika o zaistnieniu określonego zdarzenia. Powiadomienie może być wysłane drogą mailową lub poprzez dedykowany widok w aplikacji. Wyzwalaczem sygnału może być zmiana stanu obiektu lub czas.

W realizowanej aplikacji powstał moduł odpowiedzialny za zarządzanie sygnałami. W trakcie jego realizacji należało rozwiązać szereg problemów projektowych: powiadomienia dotyczące różnych rodzajów zdarzeń (incydentu, problemu, zlecenia), obsłużenie różnorodnych rodzajów sygnałów, możliwość konfiguracji treści wiadomości oraz treści tematu na podstawie rodzaju sygnału, generowanie sygnałów przy zmianie stanu, generowanie sygnałów przy upływie czasu, wysyłanie wiadomości e-mail dla każdego sygnału oraz ustalenie kategorii odbiorcy sygnału.

5.3.1 Typy sygnałów

Wymagania narzucają elastyczne podejście obsługujące obiekty różnego typu oraz sygnały różnego typu. Każdy typ sygnału dotyczy konkretnego typu obiektu: incydentu, problemu, zlecenia. Typ obiektu zawiera listę atrybutów, w oparciu o które można tworzyć treść wiadomości oraz treść tematu wiadomości. Jest to bardzo elastyczne rozwiązanie ponieważ w dowolnym momencie można dodać nowy typ obiektu co nie będzie miało wpływu na resztę modułu, podobnie jest z nowym typem sygnału. Szablon wiadomości może ulec zmianie w dowolnym momencie, co korzystnie wpływa na elastyczność. Treść wiadomości jest budowana dynamicznie na podstawie atrybutów obiektu oraz szablonu wiadomości.

Przykład: Niech obiekt zawiera atrybut: **STATUS** o wartości: **rozwiążany 5.1**.

Tabela 5.1: Treść wiadomości na podstawie szablonu

Szablon	Treść wiadomości
Incydent zmienił status na: \${STATUS}.	Incydent zmienił status na: rozwiążany.

Każdy typ sygnału ma przypisaną rangę określającą wagę sygnału:

- informacja;
- ostrzeżenie;
- błąd;

5.3.2 Rodzaj odbiorcy

Pracownik, dla którego generowany jest sygnał, określany jest na podstawie powiązania z danym zdarzeniem. Została utworzona lista kategorii odbiorców jednoznacznie wyznaczająca pracownika dla danego zdarzenia. W dowolnej chwili istnieje możliwość dodania nowej kategorii co świadczy o elastyczności podejścia.

Przykładowe rodzaje odbiorców:

- serwisant incydentu;
- menedżer grupy wsparcia;
- menedżer serwisanta incydentu;
- zgłaszający incydent;
- menedżer zgłaszającego incydent;

5.3.3 Generacja sygnałów przy zmianie stanu

Do tego celu zostały wykorzystane wyzwalacze w bazie danych. Stworzono wyzwalacz na poziomie wiersza, który po operacjach `insert` oraz `update` dodaje nowe sygnały. W tym miejscu powstał problem pobierania danych z aktualnie modyfikowanej tabeli. Rozwiążanie tej sytuacji polega na wstawieniu przez wyzwalacz identyfikatorów zmienionych wierszy do tymczasowej tabeli. Następnie wyzwalacz na poziomie tabeli, dla każdego wiersza dodaje nowy sygnał.

5.3.4 Generacja sygnałów po upływie czasu

Do realizacji tego celu zostały stworzone generatorы sygnałów. Jest to byt dodający sygnały określonego typu dla obiektów spełniających zadane kryterium. Kryterium jest budowane na podstawie czasu.

Przykład Zbliżający się termin rozwiązania incydentu. W takim wypadku kryterium wygląda następująco:

Listing 5.1: Przykładowe kryterium generatora sygnałów

```
1 SYSDATE < TARGET_RESOLUTION_DATE  
2 AND TARGET_RESOLUTION_DATE < SYSDATE + 4/24
```

W aplikacji jest jedno zadanie pełniące rolę planisty. Co pewien kwant czasu sprawdza ono, czy istnieją generatory gotowe do uruchomienia. Wybór jednego planisty zamiast oddzielnego zadania dla każdego generatora umożliwia dynamiczne zarządzanie generatorami poprzez aplikację.

5.3.5 Konstrukcja zapytania dla generatora

Każdy generator konstruuje zapytanie odnajdujące obiekty, dla których zostanie wygenerowany sygnał. Postać zapytania została pokazana na wydruku 5.2.

Listing 5.2: Szablon zapytania dla generatora

```
1 SELECT o.* , e.employee_id  
2 FROM objectType.viewName o receiverType.fromStatement  
3 WHERE criterion AND checkDuplicateCondition
```

objectType.viewName

`objectType` jest to typ obiektu, którym może być: incydent, problem lub zlecenie. Każdy typ obiektu jest związany z perspektywą w bazie danych, która zawiera atrybuty tego obiektu. Czyli `objectType.viewName` oznacza nazwę perspektywy zawierającej dane o obiektach konkretnego typu.

receiverType.fromStatement

Każdy generator ma przypisanego adresata, co zostało opisane w sekcji 5.3.2. Zmienna **receiverType** określa właśnie rodzaj odbiorcy. Każdy rodzaj odbiorcy posiada atrybut **fromStatement**, który wskazuje wiersz z tabeli EMPLOYEES, dla którego zostanie wygenerowany sygnał.

Listing 5.3: Wartość atrybutu fromStatement dla serwisanta

```
1 JOIN employees e ON e.employee_id = o.staff_member_id
```

Listing 5.4: Wartość atrybutu fromStatement dla menedżera grupy wsparcia

```
1 JOIN support_groups sg ON sg.group_id = o.group_id
2 JOIN employees e ON e.employee_id = sg.manager_id
```

criterion

Jest to kryterium, które muszą spełniać obiekty, przykładowe kryterium zostało pokazane w sekcji 5.3.4.

checkDuplicateCondition

Jest to dodatkowy warunek, który nie dopuszcza do powstawania duplikatów.

Listing 5.5: Przykładowe zapytanie dla generatora

```
1 SELECT o.* , e.employee_id
2 FROM
3     incidents_v o
4     JOIN employees e
5         ON e.employee_id = o.staff_member_id
6 WHERE
7     sysdate < target_resolution_date
8     AND target_resolution_date < sysdate + 4/24
9     AND not exists (
10         SELECT *
11         FROM signals
12         WHERE
13             type_code = '1'
14             AND object_id = o.incident_id
15     )
```

5.3.6 Wysyłanie wiadomości e-mail

Rozwiążanie tego problemu to utworzenie zadania pobierającego co pewien okres czasu sygnały dla których jeszcze nie wysłano wiadomości e-mail. Po wysłaniu wiadomości należy oznakować taki sygnał, aby uniknąć ponownego działania.

Rozdział 6

Implementacja

Rozdział ten zawiera opis implementacji fragmentów systemu Service Desk. Na wstępie zostały zaprezentowane wykorzystane technologie. Przedstawiono również narzędzia zastosowane w procesie wytwarzania oprogramowania. Na końcu zostały omówione pakiety aplikacji.

6.1 Technologie

Autor postanowił wykonać system jako aplikację WWW w technologii J2EE z interfejsem użytkownika w postaci przeglądarki. Takie rozwiązanie pozwoli na zminimalizowanie kosztów wdrożenia i aktualizacji systemu, a także umożliwi jego skalowanie w zależności od potrzeb. Ponadto pozwoli to na zdalny dostęp do zasobów systemu poza stanowiskiem pracy, co może być ważne np. dla kadry menedżerskiej i specjalistów grup wsparcia.

Spring framework

Tworzenie aplikacji WWW przy pomocy standardowej biblioteki Java jest zajęciem trudnym. Każdy twórca jest zmuszony do rozwiązania podstawowych problemów samodzielnie. Autor zdecydował się na skorzystanie z gotowego szkieletu Spring Framework. Głównym powodem jest wsparcie dla wzorca IoC¹, poprzez wstrzykiwanie zależności². Odpowiedzialność za tworzenie powiązań pomiędzy obiektami jest przeniesiona do kontenera IoC. Jest to odwrócenie sterowania w sensie tworzenia oraz wiązania obiektów. W Spring Framework kontener umożliwia używanie aspektów, więc można dodać obsługę transakcji, logowanie oraz zarządzanie bezpieczeństwem bez ingerencji w istniejący kod. Użycie tej tech-

¹IoC - Inversion of Control

²DI - Dependency Injection

niki wpływa pozytywnie na łatwość testowania obiektów, ponieważ powiązania pomiędzy obiektami są luźne [8].

Hibernate

Warstwa dostępu do danych została stworzona w technologii Hibernate. Jest to framework do realizacji dostępu do danych. Zapewnia translację danych pomiędzy relacyjną bazą danych a światem obiektowym [11], [10] Mapowanie relacyjno-obiektowe zostało utworzone przy pomocy adnotacji JPA³ (a nie plików XML). Daje to możliwość zamiany technologii Hibernate na inną technologię obsługującą standard JPA, przykładowo Oracle TopLink.

Spring Security

Warstwa bezpieczeństwa została utworzona za pomocą biblioteki Spring Security. Głównym powodem takiego wyboru jest łatwość integracji tej biblioteki ze Spring Framework oraz bogata funkcjonalność [1]. Warstwa bezpieczeństwa została zrealizowana jako osobny aspekt, co umożliwia rozdzielenie bezpieczeństwa od logiki biznesowej oraz pozostałych warstw.

Spring Web Services

Do implementacji usług sieciowych⁴ wykorzystano bibliotekę Spring Web Services. W tym przypadku powodem jest łatwość integracji z Spring Framework, ponieważ są to produkty tej samej firmy. Kolejną zaletą tej biblioteki jest podejście do tworzenia usług Contract First. Usługę sieciową rozpoczyna się od utworzenia WSDL⁵, a następnie kodu w języku wysokiego poziomu implementującego ten kontrakt. Istnieje drugie podejście Contract Last. Polega ono na wygenerowaniu kontraktu z kodu Javy. Contract First w porównaniu do Contract Last cechuje się lepszą stabilnością, w tym drugim, przy wprowadzaniu zmian, może dojść do zmiany kontraktu, co ma negatywny wpływ na klientów usługi [19].

JasperReports

Do implementacji raportów wykorzystano JasperReports. Jest to biblioteka stworzona w technologii Java umożliwiająca elastyczne tworzenie raportów w sposób niezależny od źródła danych oraz od formatu wynikowego. Skorzystano z narzędzia iReport, które umożliwia zarządzanie procesem tworzenia raportów przy pomocy graficznego interfejsu użytkownika [7]. Przypomina to tworzenie interfejsu użytkownika przy pomocy narzędzi typu RAD⁶. Twórca raportu wybiera

³JPA – Java Persistence Annotations

⁴Usługa sieciowa – (ang. Web Service)

⁵WSDL – Web Services Description Language

⁶RAD – Rapid Application Development

elementy z palety oraz przenosi je na formatkę projektu. Następnie przy pomocy widoku właściwości ustawia parametry obiektów [5]. Ważnym powodem wyboru tej technologii jest łatwość integracji ze Spring Framework. Dostępna jest klasa reprezentująca widok będący raportem wykonanym w JasperReports.

jQuery

Przy implementacji warstwy prezentacji wykorzystano bibliotekę Open Source jQuery. Jest to biblioteka ułatwiająca tworzenie dynamicznych stron przy pomocy JavaScript [12]. Skorzystano z następujących wtyczek (ang. plugin):

- tablica z sortowaniem[4];
- kontrolka do wyboru daty[21];
- zakładki[21];
- kontrolka w postaci drzewa[22];
- rozwijane menu[2];

6.2 Wykorzystane narzędzia

Rational Software Architect

Wykorzystany do utworzenia diagramów UML wstępnego projektu, który został załączony na płycie CD.

JDeveloper

Wykorzystany do utworzenia diagramów tabel oraz wygenerowania kodu SQL. Projekt JDeveloper oraz wygenerowane skrypty zostały załączone na płycie CD.

SpringSource Tool Suite

Wykorzystany jako IDE do stworzenia aplikacji w Spring framework.

Oracle Database 10g Express Edition

Wykorzystana jako relacyjna baza danych.

Apache Tomcat

Kontener serwletów.

iReport

Do utworzenia szablonów raportów JasperReports.

6.3 Pakiety aplikacji

Niniejsza sekcja zawiera omówienie pakietów aplikacji. Główny pakiet aplikacji to **sd**. W pierwszej kolejności pakiety zostały podzielone wg wybranych procesów: pakiet **im** - proces zarządzania incydentami, pakiet **pm** - proces zarządzania problemami, pakiet **em** - proces zarządzania zdarzeniami, pakiet **rf** - proces zarządzania zleceniami. Na drugim poziomie pakiety podzielono wg warstw aplikacji, które zostały omówione w sekcji 5.2.1:

dao – warstwa dostępu do danych;

domain – warstwa domenowa;

web – kontrolery stron;

service – warstwa usługowa;

validator – warstwa walidacji;

Występują również pakiety **editor**, które są odpowiedzialne za dokonanie przekształcenia parametrów żądania HTTP na obiekt wybranej klasy. Przykładowo, może to być przekształcenie identyfikatora kategorii incydentów na obiekt klasy **IncidentCategory**.

6.3.1 sd.im

Rysunek 6.1 przedstawia diagram pakietów odpowiedzialnych za proces zarządzania incydentami.

sd.im.dao

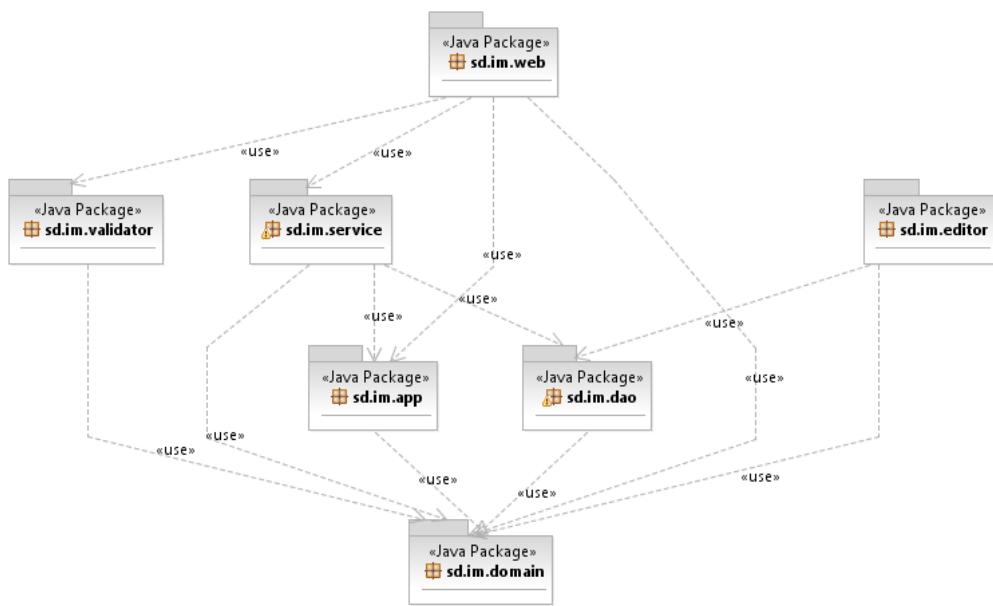
W tym pakiecie znajdują się klasy dostępu do danych odpowiedzialne za pobieranie oraz zapisywanie do bazy danych obiektów domenowych.

sd.im.domain

W tym pakiecie są klasy domenowe związane z incydentami. Główną klasą jest **Incident**, pozostałe klasy to słowniki: priorytet, kategoria, wpływ, status oraz pilność incydentu.

sd.im.app

W tym pakiecie jest klasa reprezentująca kryterium wyszukiwania incydentów.



Rysunek 6.1: Diagram pakietów dla zarządzania incydentami

sd.im.web

Jest to pakiet pełniący rolę kontrolera. Wykorzystuje usługi z pakietu **sd.im.service**. Dokonuje konfiguracji warstwy prezentacji. Korzysta z pakietu **sd.im.validator** przy walidacji.

sd.im.service

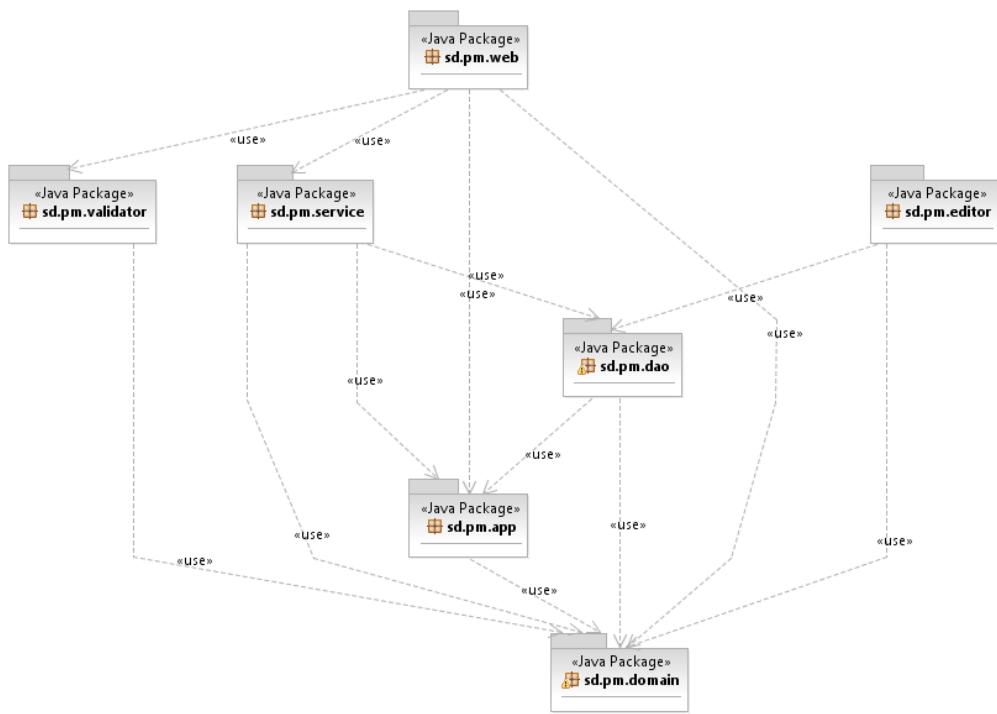
Pakiet zawierający klasy usługowe. Wykorzystuje on pakiet **sd.pm.dao** do operacji w bazie danych.

Zawiera następujące usługi:

- dodanie incydentu,
- edycja incydentu,
- dodanie komentarza dla incydentu,
- pobranie wszystkich wartości danego słownika.

6.3.2 sd.pm

Rysunek 6.2 przedstawia diagram pakietów odpowiedzialnych za proces zarządzania problemami.



Rysunek 6.2: Diagram pakietów dla zarządzania problemami

sd.pm.dao

Pakiet odpowiedzialny za pobieranie oraz zapisywanie do bazy danych obiektów domenowych z pakietu **sd.pm.domain**.

sd.pm.domain

Pakiet zawierający obiekty domenowe związane z problemami. Główną klasą jest **Problem**. Pozostałe klasy reprezentują słowniki: kategoria, status, priorytet, wpływ oraz pilność.

sd.pm.app

Pakiet zawierający klasy reprezentującą kryterium wyszukiwania problemów.

sd.pm.web

Jest to pakiet pełniący rolę kontrolera. Wykorzystuje usługi z pakietu **sd.pm.service**. Dokonuje konfiguracji warstwy prezentacji. Korzysta z pakietu **sd.pm.validator** przy walidacji.

sd.pm.service

Pakiet zawierający klasy usługowe. Wykorzystuje on pakiet sd.pm.dao do operacji w bazie danych.

Zawiera następujące usługi:

- dodanie problemu,
- edycja problemu,
- dodanie komentarza dla problemu,
- pobranie wszystkich wartości danego słownika.

sd.pm.validator

Pakiet odpowiedzialny za sprawdzanie poprawności przy zapisywaniu problemów.

sd.pm.editor

Jest to pakiet, który dokonuje przekształcenia napisu na obiekt konkretnej klasy. Dla każdej klasy słownikowej istnieje klasa edytora, która przekształca napis reprezentujący identyfikator na konkretny obiekt. Pakiet jest wykorzystywany do zamiany parametrów żądania HTTP na obiekt wybranej klasy, przykładowo umożliwia to zamianę identyfikatora kategorii problemu na obiekt klasy **ProblemCategory**. Jest to eleganckie podejście, ponieważ umożliwia to operowanie na obiektach, a nie na napisach.

6.3.3 sd.em

Rysunek 6.3 przedstawia diagram pakietów odpowiedzialnych za proces zarządzania zdarzeniami.

sd.em.dao

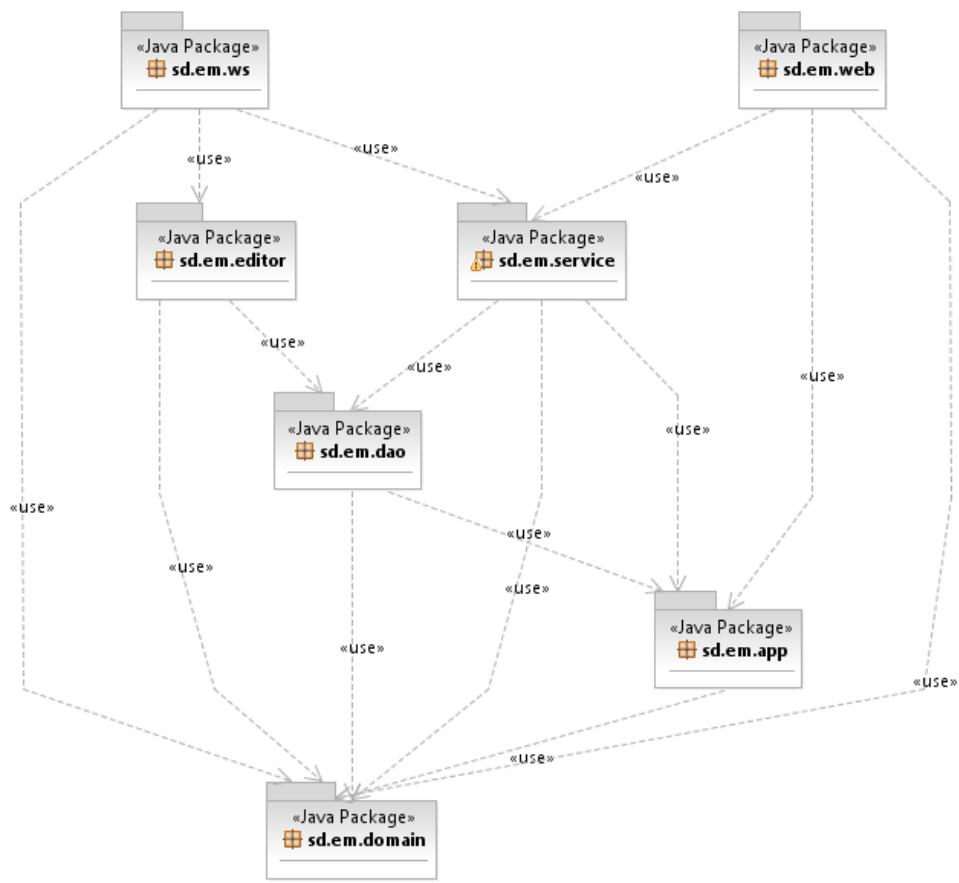
Pakiet odpowiedzialny za pobieranie oraz zapisywanie do bazy danych obiektów domenowych z pakietu sd.em.domain.

sd.em.domain

W tym pakiecie są klasy domenowe związane ze zdarzeniami. Główną klasą jest Event, pozostałe klasy to słowniki: kategoria, status oraz ranga.

sd.em.app

W tym pakiecie jest klasa reprezentująca kryterium wyszukiwania zdarzeń.



Rysunek 6.3: Diagram pakietów dla zarządzania zdarzeniami

sd.em.web

Jest to pakiet pełniący rolę kontrolera. Wykorzystuje usługi z pakietu sd.em.service. Dokonuje konfiguracji warstwy prezentacji. Korzysta z pakietu sd.em.validator przy walidacji.

sd.em.service

Pakiet zawierający klasy usługowe. Wykorzystuje on pakiet sd.em.dao do operacji w bazie danych.

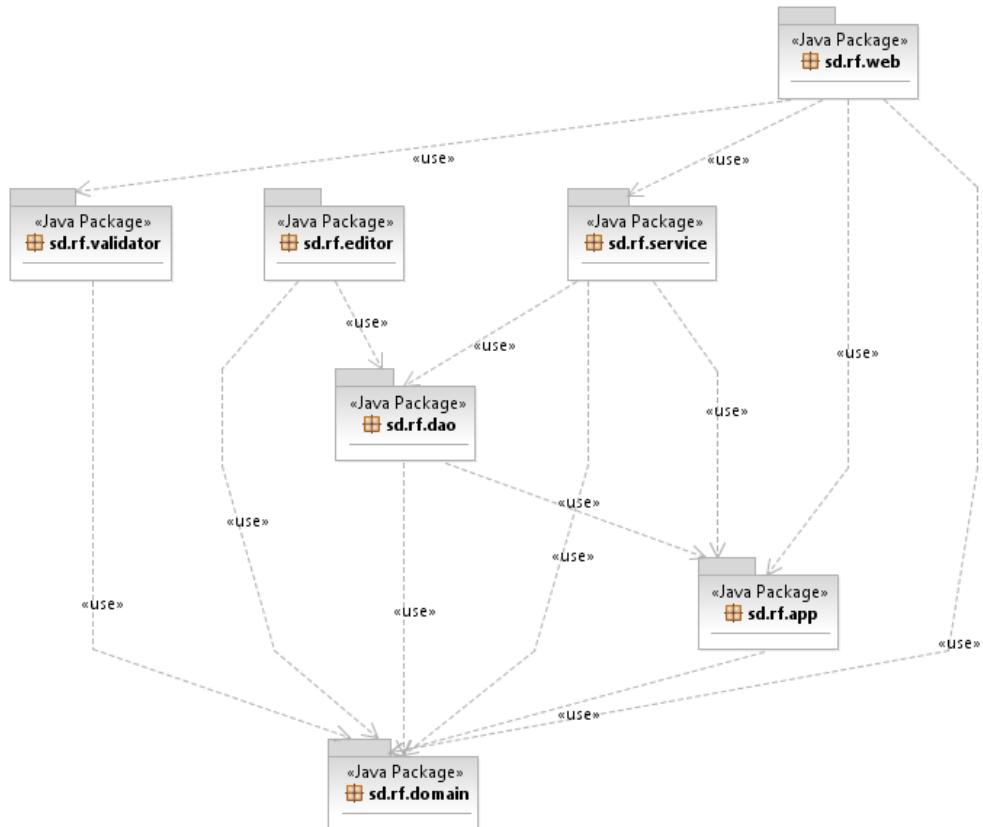
Zawiera następujące usługi:

- dodanie zdarzenia,
- edycja zdarzenia,
- dodanie komentarza dla zdarzenia,

- pobranie wszystkich wartości danego słownika.

6.3.4 sd.rf

Rysunek 6.4 przedstawia diagram pakietów odpowiedzialnych za proces zarządzania zleceniami.



Rysunek 6.4: Diagram pakietów dla zarządzania zleceniami

sd.rf.dao

Pakiet odpowiedzialny za pobieranie oraz zapisywanie do bazy danych obiektów domenowych z pakietu `sd.rf.domain`.

sd.rf.domain

W tym pakiecie są klasy domenowe związane ze zleceniami. Główną klasą jest `ServiceRequest`, pozostałe klasy to słowniki: kategoria, status oraz priorytet.

sd.rf.app

Pakiet zawierający kryterium wyszukiwania zleceń.

sd.rf.web

Jest to pakiet pełniący rolę kontrolera. Wykorzystuje usługi z pakietu sd.rf.service. Dokonuje konfiguracji warstwy prezentacji. Korzysta z pakietu sd.rf.validator przy walidacji.

sd.rf.service

Pakiet zawierający klasy usługowe. Wykorzystuje on pakiet sd.rf.dao do operacji w bazie danych.

Zawiera następujące usługi:

- dodanie zlecenia,
- edycja zlecenia,
- dodanie komentarza dla zlecenia,
- pobranie wszystkich wartości danego słownika.

6.3.5 sd.signal

Rysunek 6.5 przedstawia diagram pakietów odpowiedzialnych za zarządzanie sygnałami.

sd.signal.dao

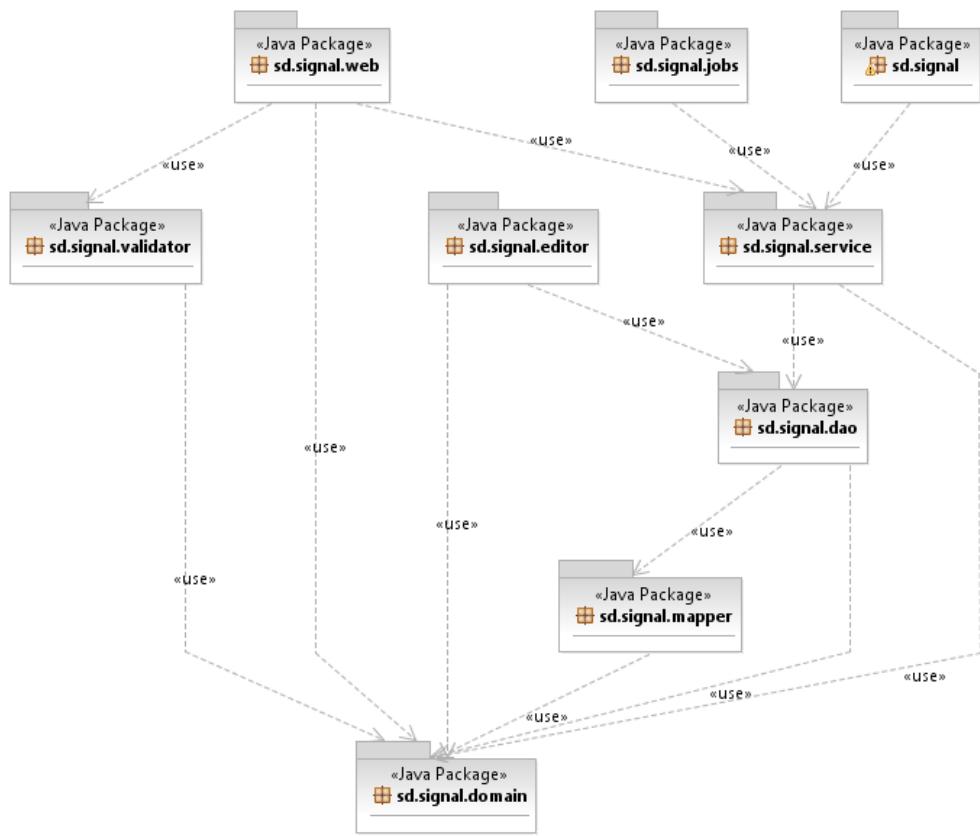
Pakiet odpowiedzialny za pobieranie oraz zapisywanie do bazy danych obiektów domenowych z pakietu sd.signal.domain.

sd.signal.domain

Pakiet zawierający klasy domenowe reprezentujące sygnał oraz generator sygnałów.

sd.signal.web

Jest to pakiet pełniący rolę kontrolera. Wykorzystuje usługi z pakietu sd.signal.service. Dokonuje konfiguracji warstwy prezentacji. Korzysta z pakietu sd.signal.validator przy walidacji.



Rysunek 6.5: Diagram pakietów dla zarządzania sygnałami

sd.signal.service

Pakiet zawierający klasy usługowe. Wykorzystuje on pakiet `sd.rf.dao` do operacji w bazie danych.

Zawiera następujące usługi:

- dodanie typu sygnału,
- edycja typu sygnału,
- dodanie generatora sygnałów,
- edycja generatora sygnałów,
- pobranie sygnałów dla danego generatora,
- pobranie wszystkich wartości danego słownika.

sd.signal.jobs

Pakiet zawierający klasy reprezentujące zadania (job). Pierwsze zadanie odpowiedzialne jest za zarządzanie generatorami sygnałów. Sprawdza ono co pewien kwant czasu, czy istnieje gotowy generator. Drugie zadanie odpowiedzialne jest za wysyłanie wiadomości e-mail dla sygnałów.

sd.signal.mapper

Jest to pakiet dokonujący przekształcenia wiersza z rezultatu zapytania na obiekt klasy Signal. Jest to wykorzystywane przy dostępie do bazy danych przy pomocy biblioteki JDBC.

Rozdział 7

Wsparcie systemu dla firmy Red Host S.A.

W tym rozdziale zostanie przedstawiona zrealizowana aplikacja i przykładowe scenariusze jej wykorzystania.

7.1 Scenariusz użycia na poziomie biznesowym

Sekcja ta zwiera przykładowy scenariusz na poziomie biznesowym. Przedstawiono cykl życia incydentu od momentu zgłoszenia przez klienta korzystającego z usługi do momentu zamknięcia incydentu.

1. Klient zgłasza nowy incydent.
2. Pracownik pierwszej linii sprawdza nieprzypisane incydenty oraz dokonuje przypisania incydentu do własnej osoby.
3. Pracownik pierwszej linii ustawia podstawowe dane dotyczące incydentu. Podejmuje próbę znalezienia rozwiązania, wykorzystując KEDB. Nie znajduje rozwiązania. Po pewnym czasie zostaje wysłane powiadomienie do menedżera incydentów o zbliżającym się terminie rozwiązania incydentu.
4. Menedżer incydentów dokonuje eskalacji incydentów do pracownika drugiej linii.
5. Pracownik drugiej linii zostaje powiadomiony drogą mailową o nowo przypisanym incydencie.
6. Pracownik drugiej linii dokonuje analizy incydentu wykorzystując w tym celu CMDB. Po dokonaniu analizy stwierdza, że incydent jest związany z istnieniem problemu, więc dodaje nowy problem.

7. W trakcie analizy oraz diagnozy problemu zostaje znalezione rozwiązanie tymczasowe oraz rozwiązanie stałe.
8. Menedżer problemów dodaje nowy znany błąd.
9. Rozwiązanie tymczasowe zostaje wykorzystane do rozwiązania incydentu. Pracownik drugiej linii zmienia status incydentu na rozwiązany.
10. Pracownik pierwszej linii wysyła zapytanie do klienta, czy incydent został pomyślnie rozwiązany. Po otrzymaniu pozytywnej odpowiedzi dokonuje zamknięcia incydentu.
11. W celu rozwiązania problemu należy dokonać zmiany w infrastrukturze. Zostaje zgłoszone RFC.

7.2 Scenariusz użycia na poziomie systemowym

Na podstawie scenariusza biznesowego przedstawiono scenariusz użycia na poziomie systemowym.

7.2.1 Zgłoszenie incydentu

W sytuacji naruszenia poziomu usług w umowie SLA, klient ma możliwość zgłoszenia incydentu w aplikacji poprzez formularz WWW (rysunek: 7.1). Klient musi podać temat incydentu oraz opisać szczegółowo co się wydarzyło.

Incydent	
Temat:	<input type="text"/>
Opis:	<input type="text"/>

Rysunek 7.1: Formularz nowego incydentu

Klient może również zgłosić incydent drogą telefoniczną lub mailową. W takim przypadku pracownik pierwszej linii doda nowy incydent w imieniu klienta. Czynność ta jest bardzo podobna do zgłoszenia incydentu bezpośrednio przez klienta, różnica jest taka, że pracownik musi wyszukać klienta w bazie oraz wybrać źródło: e-mail lub telefon. Dodatkowe dane do wypełnienia zostały pokazane na rysunku: 7.2.

Zgłoszone przez:	Korczak Kamil
Źródło:	Phone call

Rysunek 7.2: Formularz nowego incydentu - szczegóły

7.2.2 Przypisanie incydentu

W zrealizowanym systemie pracownik pierwszej linii dokonuje przypisania do incydentu samodzielnie. Pracownik pierwszej linii ma dostępną funkcjonalność podeerzenia nieprzypisanych incydentów. Po wybraniu incydentu zostanie przeniesiony do karty incydentu. Karta incydentu jest najważniejszym widokiem związanym z zarządzaniem incydentami. W tym miejscu zawarte są szczegółowe dane dotyczące incydentu. Dostępna jest również możliwość przypisania serwisanta do incydentu, co widoczne jest na rysunku: 7.3.

Usługa:	Free Hosting	Szczegóły	
Priorytet:	--Select--	Status:	Open
Wpływ:	--Select--	Pilnośc:	--Select--
Kategoria:		Źródło:	E-Mail
Serwisant:	1_LINE 1	Grupa wsparcia:	--Select--

Rysunek 7.3: Karta incydentu - przypisanie serwisanta

Takie podejście może spowodować, że nie wszystkie incydenty zostaną przypisane w ustalonym czasie. Aby zapobiec takiej sytuacji istnieje możliwość utworzenia generatora sygnałów 5.3.4, który będzie odpowiedzialny za powiadomienie menedżera incydentów o wszystkich nieprzypisanych incydentach w ustalonym czasie. W przykładowych danych zostały skonfigurowane dwa generatory: pierwszy powiadamia menedżera incydentów o nieprzypisanych incydentach po dwóch godzinach od czasu zgłoszenia incydentu, drugi po czterech godzinach. W takim podejściu należy również pamiętać o systemie wynagrodzeń dla pracowników pierwszej linii. Wysokość premii powinna zależeć od liczby odebranych incydentów. Aplikacja umożliwia wygenerowanie raportu o liczbie incydentów przypisanych do danej grupy/pracownika w podanym okresie czasu.

7.2.3 Karta incydentu

Rysunek 7.4 przedstawia obszar na karcie incydentu zawierający podstawowe informacje o incydencie, czyli: identyfikator, priorytet, aktualny status, przypisany serwisant, data zgłoszenia, termin rozwiązania, data rozwiązania, data zamknięcia, temat oraz opis.

ID:	1150	Priorytet:	Medium	Status:	Open	Zgłoszone przez:	2_LINE 2
Data zgłoszenia:	2010-07-11 17:04	Termin rozwiązania:	2010-07-12 01:04	Data rozwiązania:		Data zamknięcia:	
Temat: e-mail nie działa							
Opis: W trakcie wysyłania wiadomości maila wyskakuje mi okienko z komunikatem błędu:							

Rysunek 7.4: Karta incydentu - dane podstawowe

Poniżej umiejscowione są szczegóły. Obszar ten podlega edycji. Najważniejszym elementem jest usługa przypisana do incydentu. Na liście usług do wyboru są tylko usługi dostępne dla klienta, który zgłosił incydent. W tym miejscu istnieje możliwość dokonania zmiany niektórych atrybutów zawartych w obszarze podstawowym. Dodatkowo można przypisać kategorie incydentu, wybrać grupę wsparcia oraz wpisać rozwiązanie, co jest pokazane na rysunku: 7.5.

Usługa:	-Select- Szczegóły		
Priorytet:	Medium	Status:	Open
Wpływ:	Affects Department	Pilność:	Low
Kategoria	Client card	Źródło:	Phone call
Serwisant:	1_LINE 1	Szczegóły	Grupa wsparcia:
Rozwiązanie: <input type="text"/>			

Rysunek 7.5: Karta incydentu - szczegóły

7.2.4 Eskalacja

Po upływie określonego czasu nie rozwiązany incydent powinien zostać eskalowany do pracownika drugiej linii. W aplikacji zostały utworzone generatory sygnałów, co zostało szczegółowo opisane w sekcji 5.3. Istnieje możliwość skonfigurowania zasad powiadamiania odpowiednich pracowników. W tym przypadku

należy poinformować menedżera incydentów o konieczności przekierowania sygnału do pracownika drugiej linii. Istnieje również możliwość poinformowania bezpośredniego przełożonego pracownika pierwszej linii. Menedżer po otrzymaniu wiadomości e-mail dokonuje przypisania pracownika drugiej linii, w taki sam sposób jak to zrobił pracownik pierwszej linii, zostało to przedstawione na rysunku: 7.3. Sygnały widoczne są również w aplikacji, każdy zalogowany użytkownik na stronie głównej ma dostępną tabelę z sygnałami (rysunek: 7.6).

Sygnały				
Data zgłoszenia	Temat	Wiadomość	Ranga	
2010-05-15 20:58:09.0	Nowy incydent	Został przypisany incydent: 50, temat: USER1 2010-05-15, dodany: 10/05/15 abc	Info	
2010-05-15 20:59:56.0	Odpisano incydent	Incydent: 50, temat: USER1 2010-05-15 został przypisany innej osobie.	Info	

Rysunek 7.6: Sygnały na stronie głównej

Każdy sygnał zawiera również odnośnik do karty obiektu, którego dotyczy. W tym przypadku będzie to odnośnik do karty incydentu. Umożliwia to przyspieszenie nawigacji po systemie.

7.2.5 Powiadomienie pracownika drugiej linii

W sytuacji, gdy incydent zostanie przypisany do pracownika drugiej linii, zostanie on powiadomiony drogą mailową. Do tego celu również zostały wykorzystane sygnały, co zostało opisane w sekcji: 5.3.

7.2.6 Historia incydentu

Dla każdego incydentu przechowywana jest pełna historia. Rysunek 7.7 przedstawia obszar z historią incydentu. W tym miejscu widoczna jest każda zmiana. Przy zmianie atrybutu prezentowana jest stara oraz nowa wartość. Dzięki temu, każdy serwisant zajmujący się danym incydentem ma możliwość podejrzenia wszystkich czynności dokonanych w przeszłości.

Historia	
1	2
2010-07-14 23:22:30.0, SYS SYS	
1. Kategoria: - -> Client card 2. Wpływ: - -> Affects Department 3. Priorytet: - -> Medium 4. Termin rozwiązania: - -> 2010-07-12 01:04:46.0 5. Pilność: - -> Low 6. Serwisant: - -> 1_LINE 1 7. Grupa wsparcia: - -> grupa 2	

Rysunek 7.7: Karta incydentu - historia

7.2.7 Dodanie problemu

W trakcie analizy incydentu może się okazać, że incydent związany jest z istnieniem problemu. Na karcie incydentu serwisant ma możliwość dodania nowego problemu, co widoczne jest na rysunku: 7.8.

The screenshot shows a user interface for managing incidents. At the top, there are two buttons: 'Zapisz' (Save) and 'Dodaj problem' (Add problem), with 'Dodaj problem' highlighted by a red box. Below this, the word 'Incydent' is displayed in bold blue text. Underneath is a table with the following data:

ID:	2050	Priorytet:	Status:	Open	Zgłoszone przez:	USER 2
-----	------	------------	---------	------	------------------	--------

Rysunek 7.8: Karta incydentu - dodanie problemu

Dodanie problemu w taki sposób spowoduje jego automatyczne powiązanie z incydentem. Będzie to widoczne na karcie problemu (rysunek: 7.9).

The screenshot shows a table titled 'Incydenty' with the following data:

Id	Temat	Status	Priorytet	
2050	2010 user2	Open		Usuń

Rysunek 7.9: Karta problemu - powiązane incydenty

Istnieje również możliwość przypisania nowych incydentów do problemu, jak również usunięcia przypisania, w przypadku popełnienia błędu.

7.2.8 Karta problemu

The screenshot shows a card for managing problems. At the top, it displays basic information in a table:

ID:	100	Priorytet:	Medium	Status:	On Hold
Znany błąd:	false	Data zgłoszenia:	2010-05-15 21:01	Data zamknięcia:	

Below this are two text input fields labeled 'Temat:' and 'Opis:', each with a corresponding text area for input.

Rysunek 7.10: Karta problemu - dane podstawowe

Karta problemu zawiera wszystkie dane związane z problemem. Jest to widok bardzo podobny do karty incydentu. Tutaj również dostępne są podstawowe dane dotyczące problemu: identyfikator problemu, priorytet, status, data zgłoszenia oraz data zamknięcia. Widoczne są one na rysunku 7.10.

Szczegóły			
Priorytet:	Medium	Status:	On Hold
Wpływ:	Affects Group	Pilność:	Low
Kategoria:		Znany błąd:	<input type="checkbox"/>
Serwisant:		Grupa wsparcia:	grupa 2

Rysunek 7.11: Karta problemu - dane szczegóły

Ponadto dostępne są szczegółowe informacje dotyczące problemu (rysunek 7.11). W tym miejscu można dokonać edycji niektórych danych podstawowych: priorytetu, statusu oraz danych dodatkowych: wpływu, pilności, kategorii, serwanta oraz grupy wsparcia.

7.2.9 Analiza problemu

Analiza	
Wpływ:	
Główna przyczyna:	
Symptomy:	

Rysunek 7.12: Karta problemu - analiza

Każdy problem powinien zostać dokładnie przeanalizowany, aby móc zidentyfikować jego symptomy, określić szczegółowy wpływ na biznes oraz ustalić główną przyczynę jego wystąpienia (rysunek 7.12).

7.2.10 Rozwiążanie tymczasowe

W trakcie analizy problemu może zostać znalezione rozwiązanie tymczasowe lub rozwiązanie stałe. Na karcie problemu istnieje możliwość dodania rozwiązania dla problemu, co pokazuje rysunek 7.13. Po wybraniu jednej z opcji zostanie wyświetlony formularz do uzupełnienia szczegółów rozwiązania: tematu oraz opisu.

Rozwiążanie

Rozwiążanie tymczasowe:

Rozwiążanie:

Rysunek 7.13: Karta problemu - analiza

W przypadku, gdy istnieje rozwiązanie tymczasowe, można dodać nowy znany błąd. Ta czynność może zostać wykonana tylko przez menedżera problemów, poprzez zaznaczenie opcji widocznej na rysunku 7.14.



Rysunek 7.14: Karta problemu - znany błąd

Znane błędy są dostępne w KEDB. W aplikacji jest dostępna funkcjonalność wyszukiwania znanych błędów (rysunek: 7.15).

Szukaj znanych błędów

Id	Temat	Kategoria	Symptom	Główna przyczyna	Rozwiążanie tymczasowe

Szukaj

Brak wyników

Rysunek 7.15: Wyszukiwanie znanych błędów

7.2.11 Rozwiążanie incydentu

Rozwiążanie tymczasowe znalezione w trakcie analizy problemu, może zostać wykorzystane do rozwiązania incydentu. W aplikacji, w celu rozwiązania incydentu, należy zmienić status incydentu na rozwiązany oraz podać treść rozwiązania, co jest pokazane na rysunku: 7.16.

7.2.12 Zamknięcie incydentu

Po wdrożeniu rozwiązania, należy wysłać zapytanie do klienta, czy wszystko funkcjonuje prawidłowo. Rysunek 7.17 przedstawia szczegółowe dane o zgłaszającym, między innymi znajduje się tam lista telefonów oraz adres e-mail.

Po otrzymaniu pozytywnej odpowiedzi można zamknąć incydent. Czynność ta dokonywana jest w podobny sposób jak rozwiązanie, co zostało pokazane na rysunku: 7.16, wystarczy zmienić status incydentu na zamknięty.

Szczegóły

Usługa:	-Select--	Szczegóły	
Priorytet:	Low	Status:	Resolved
Wpływ:	Affects Department	Pilność:	Urgent
Kategoria:	Explorer	Źródło:	Phone call
Serwisant:	1_LINE 1	Grupa wsparcia:	grupa 3

Rozwiążanie:

rozwiązanie

Rysunek 7.16: Karta incydentu - rozwiązanie

Szczegóły zgłaszającego

Imię:	2_LINE	Nazwisko:	2
Telefon komórkowy:	200	Telefon:	100
Email:	2_LINE@firma.com.pl		

Rysunek 7.17: Karta incydentu - szczegóły zgłaszającego

7.3 Podsumowanie

Zrealizowany fragment systemu dotyczy procesów zarządzania incydentami oraz zarządzania problemami. W ramach procesu zarządzania incydentami udało się zrealizować funkcjonalności zgłoszenia nowego incydentu, przydzielania incydentów, wyszukiwania incydentów, dodawania komentarzy, które umożliwiają komunikację pomiędzy pracownikiem technicznym a klientem usługi. Dla każdego incydentu przechowywana jest pełna historia, co umożliwia szybsze przyswojenie nowym pracownikom. Zostało zaprojektowane oraz zaimplementowane rozwiązanie dla eskalacji: sygnały. Każda organizacja ma możliwość określenia reguł eskalacji dla własnych potrzeb bez ingerencji programisty. W ramach procesu zarządzania problemami zrealizowano funkcjonalności dodawania nowych problemów, powiązania problemu z listą incydentów, dodania rozwiązania problemu (tymczasowego oraz stałego) oraz dodawania znanych błędów, które dostępne są w KEDB. Dla obu procesów zostały zrealizowane przykładowe raporty.

W zrealizowanym fragmencie systemu brakuje powiązania z procesem zarządzania konfiguracją. W trakcie analizy incydentu warto skorzystać z CMDB w celu określenia wpływu incydentu oraz zidentyfikowania przyczyny jego powstania. W systemie brakuje również możliwości zgłoszenia RFC, ponieważ rozwiązanie problemu, często związane jest z dokonaniem zmiany w infrastrukturze IT. W tej pracy zrezygnowano z wyżej wymienionych funkcjonalności, ponieważ są one tematem pracy dyplomowej inżynierskiej Adriana Wiśniewskiego.

Kolejnym problemem systemu jest automatyzacja. Nowoczesne systemy

Service Desk automatycznie przydzielają zgłoszenia dla serwisantów. Niektóre z nich uwzględniają przy tym urlopy serwisantów oraz ich obciążenie.

Rozdział 8

Testy i ocena

8.1 Środowiska testowe

System był testowany na następujących środowiskach:

Środowisko 1

System operacyjny: Windows 7 Ultimate 64 bitowy

Procesor: Intel Core(TM) i5 CPU 750 2,67 GHz

RAM: 4,00 GB

Baza danych: Oracle 10g Express Edition

Kontener serwletów: SpringSource tc Server v6.0

JDK: jdk1.6.0_17 (64 bitowa)

Środowisko 2

System operacyjny: Windows 7 Professional 64 bitowy

Procesor: Intel Core(TM) i3 CPU M350 2,27 GHz

RAM: 4,00 GB

Baza danych: Oracle 10g Express Edition

Kontener serwletów: Apache Tomcat 6.0.29

JDK: jdk1.6.0_21

8.2 Testy jednostkowe

Element	Coverage	Covered Instructions	Total Instructions
src/main/java	29,5 %	3323	11272
sd.em.editor	100,0 %	8	8
sd.em.ws	100,0 %	266	266
sd.event.service	100,0 %	62	62
sd.validator	100,0 %	12	12
sd.dictionary	92,7 %	51	55
sd.util	84,2 %	16	19
sd.im.dao	78,4 %	250	319
sd.pm.service	78,3 %	123	157
sd.pm.web	76,3 %	406	532
sd.em.service	75,0 %	57	76
sd.im.web	73,2 %	257	351
sd.pm.dao	68,8 %	190	276
sd.pm.validator	57,8 %	59	102
sd.im.validator	55,4 %	51	92
sd.signal.validator	48,7 %	73	150
sd.em.domain	46,7 %	226	484
sd.im.service	43,9 %	141	321
sd.pm.domain	41,0 %	443	1080
sd.domain	26,6 %	237	890
sd.im.domain	25,5 %	262	1029
sd.im.app	14,7 %	42	285
sd.pm.app	10,9 %	35	322
sd.dao	7,0 %	10	143
sd.signal.domain	6,3 %	46	727
sd.app	0,0 %	0	100
sd.editor	0,0 %	0	344
sd.em.app	0,0 %	0	152
sd.em.dao	0,0 %	0	21
sd.em.web	0,0 %	0	90
sd.im.editor	0,0 %	0	44
sd.pm.editor	0,0 %	0	24
sd.rf.app	0,0 %	0	161
sd.rf.dao	0,0 %	0	30
sd.rf.domain	0,0 %	0	524
sd.rf.editor	0,0 %	0	12
sd.rf.service	0,0 %	0	62
sd.rf.validator	0,0 %	0	43
sd.rf.web	0,0 %	0	230

Rysunek 8.1: Pokrycie testów jednostkowych

Do przetestowania aplikacji został stworzony przykładowy zestaw testów jednostkowych. Rysunek 8.1 pokazuje ich pokrycie. Pokrycie utworzonego zestawu testów nie jest wysokie, ponieważ są to tylko przykładowe testy jednostkowe. Dla systemów używanych w środowiskach produkcyjnych, należałoby dążyć do pełnego pokrycia.

Autor w początkowym etapie rozwoju aplikacji tworzył ją zgodnie z techniką

TDD¹. Jest to najlepsza technika do tworzenia testów jednostkowych, ponieważ powinny one powstawać równolegle z kodem aplikacji. W przypadku tej techniki powstają nawet przed utworzeniem kodu.

Autor zrezygnował z utworzenia pełnego zestawu testów jednostkowych, ponieważ zrealizowany system nie będzie wykorzystywany w środowisku produkcyjnym, nie będzie potrzeby pielęgnacji oraz utrzymywania systemu, a testy jednostkowe tworzone są głównie po to, aby zmniejszyć koszty utrzymania systemu, ułatwić proces refaktoryzacji oraz wprowadzania nowych funkcjonalności. Dodatkowo same testy jednostkowe nie gwarantują działania systemu. Do przetestowania systemu lepsze są testy funkcjonalne, które zostały omówione w następnej sekcji.

Testy jednostkowe umożliwiły wykrycie drobnych błędów, które zostały zidentyfikowane szybciej niż przy pomocy testów funkcjonalnych. Po pewnym czasie rozwoju aplikacji zdecydowanie więcej błędów zostało wykrytych przy pomocy testów funkcjonalnych, co było kolejnym powodem zaprzestania dalszego rozwoju zestawu testów jednostkowych.

8.3 Przykładowe testy funkcjonalne

8.3.1 Cykl życia incydentu

Cel

Test sprawdza standardowy cykl życia incydentu od momentu zgłoszenia przez klienta dodania do momentu dokonania rozwiązania.

Kroki

1. Zaloguj się na użytkownika USER1/USER1.
2. Dodaj nowy incydent.

Punkty weryfikacyjne:

- Zostanie wyświetlony komunikat: *Dodano pomyślnie incydent.*

3. Wyloguj się i zaloguj na użytkownika 1_LINE/1_LINE.
4. Wybierz opcje Zarządzanie incydentami->Nie przypisane incydenty.
5. Kliknij w temat dodanego incydentu.

Punkty weryfikacyjne:

- Zostanie wyświetlona karta incydentu.

¹TDD – Test-driven development

6. Przypisz 1_LINE jako serwisanta.

Punkty weryfikacyjne:

- Zostanie wyświetlony komunikat: *Zapisano pomyślnie incydent.*

7. Ustaw priorytet, wpływ oraz pilność. Dokonaj zapisania.

Punkty weryfikacyjne:

- Zostanie wyświetlony komunikat: *Zapisano pomyślnie incydent.*

8. Przypisz 2_LINE jako serwisanta.

Punkty weryfikacyjne:

- Zostanie wyświetlony komunikat: *Zapisano pomyślnie incydent.*

9. Przejdź na stronę główną.

Punkty weryfikacyjne:

- Dla dodanego incydentu powinny zostać wygenerowane dwa sygnały:
 - o przypisaniu incydentu oraz o odpisaniu.

10. Wyloguj się i zaloguj na użytkownika 2_LINE/2_LINE.

Punkty weryfikacyjne:

- Dla dodanego incydentu powinien zostać wygenerowany jeden sygnał
 - o przypisaniu incydentu.

11. Przejdź na kartę dodanego incydentu, poprzez kliknięcie w ostatnią kolumnę w wygenerowanym sygnale.

12. Dodaj nowy problem.

Punkty weryfikacyjne:

- Zostanie wyświetlony komunikat: *Dodano pomyślnie problem.*

13. Wyloguj się i zaloguj na użytkownika PBR_MAN/PBR_MAN.

14. Wyszukaj dodany problem w zakładce: Zarządzanie problemami->Szukaj problemów.

15. Przejdź na kartę problemu.

16. Dodaj rozwiązanie tymczasowe.

Punkty weryfikacyjne:

- Zostanie wyświetlony komunikat: *Dodano rozwiązanie.*

17. Dodaj nowy znany błąd, poprzez zaznaczenie checkbox-a: **Znany błąd**.

Punkty weryfikacyjne:

- Zostanie wyświetlony komunikat: *Zapisano pomyślnie problem.*

18. Wyloguj się i zaloguj na użytkownika **2_LINE/2_LINE**.

19. Wybierz opcje **Zarządzanie problemami->Baza znanych błędów**.

20. Wyszukaj dodany znany błąd. Skopiuj rozwiązanie błędu.

21. Przejdź na kartę incydentu. Wklej skopiowane rozwiązanie w pole: **Rozwiązanie**.
Zmień status incydentu na rozwiązyany. Dokonaj zapisania.

Punkty weryfikacyjne:

- Zostanie wyświetlony komunikat: *Zapisano pomyślnie incydent.*
- Została zmieniona data rozwiązania na bieżącą datę.

8.3.2 Ocena testów funkcjonalnych

Testy funkcjonalne umożliwiły zidentyfikowanie błędów, które nie zostały wykryte podczas testów jednostkowych. Aplikacja była testowana w trakcie powstawania. Większość błędów została wykryta w krótkim okresie czasu licząc od momentu ich pojawienia się, co miało pozytywny wpływ na czas ich rozwiązania. Podczas większych zmian systemu powstawały dodatkowe błędy, ale dzięki zestawowi standardowych testów funkcjonalnych były szybko wykrywane.

Wszystkie testy były przeprowadzane ręcznie, co niestety jest czasochłonne. Do testowania aplikacji warto wykorzystać narzędzie umożliwiające testowanie automatyczne. Przykładem takiego narzędzia jest IBM Rational Functional Tester.

8.4 Testy wydajnościowe

8.4.1 Metoda mierzenia czasu

Do mierzenia czasu został utworzony Handler Interceptor. Jest to klasa implementująca interfejs `org.springframework.web.server.HandlerInterceptor`. Interfejs zawiera trzy metody. Do mierzenia czasu zostały wykorzystane dwie. Metoda `preHandle` wywoływana jest przed obsłużeniem żądania, natomiast metoda `afterCompletion` po zrenderowaniu widoku [13]. Takie podejście umożliwia zmierzenie czasu żądania, od momentu otrzymania żądania na serwerze, po zrenderowanie widoku. Jest to w przybliżeniu czas obsłużenia żądania na serwerze. Pominięte są czynności wykonywane przez kontener serwletów przed i po żądaniu oraz czynności wykonywane przez kontener IoC, które w małym stopniu

wpływają na czas żądania. Wynika z tego, że takie podejście do mierzenia czasu jest poprawne, ponieważ mierzona jest zdecydowana większość czasu spędzonego na serwerze. Aczkolwiek pomijany jest czas transportu oraz czas zrenderowania widoku w przeglądarce. Oba czasy oczywiście są odczuwalne przez klienta, ale nie wchodzą w skład systemu. Czas zrenderowania widoku zależy od wykorzystywanej przeglądarki przez użytkownika oraz od mocy obliczeniowej stacji roboczej użytkownika, natomiast czas transportu zależy od sieci. Czas zrenderowania widoku na środowisku pierwszym waha się w okolicach 10 ms.

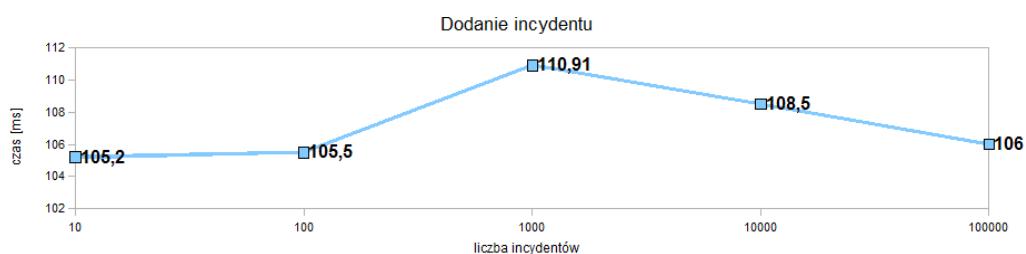
Taka metoda mierzenia czasu ma zaletę łatwości dodania do istniejącego kodu. Nie trzeba dokonywać zmian. Cały algorytm mierzenia jest zawarty w jednej klasie. Wystarczy w pliku konfiguracyjnym dodać nowy Interceptor, co zajmuje tylko jedną linijkę kodu. Nie trzeba również dokonywać ponownej komplikacji kodu, co również jest ogromną zaletą.

8.4.2 Testy wydajnościowe

Jest to rodzaj testów polegających na badaniu czasu odpowiedzi krytycznych dla biznesu funkcji. Kryterium testów jest dokonanie sprawdzenia, czy funkcje wykonywane są w akceptowalnym czasie. Zostały przetestowane dwie funkcjonalności: dodawania incydentów i wyszukiwania incydentów, ponieważ są to najczęściej wykonywane czynności w systemie Service Desk.

Każdy z testów przeprowadzono wielokrotnie, a następnie wyliczono wartość średnią.

Dodanie incydentu



Rysunek 8.2: Czas dodania incydentu w zależności od liczby incydentów w bazie danych

Rysunek 8.2 przedstawia czas dodania incydentu w zależności od liczby incydentów w bazie danych. Zgodnie z oczekiwaniami czas ten praktycznie nie zależy od liczby incydentów. Średni czas nie przekracza 120 ms, co jest akceptowalną wartością dla użytkownika końcowego.

Wyszukiwanie incydentów

Rysunek 8.3 przedstawia czas wyszukiwania incydentów w zależności od liczby incydentów w bazie danych. Zgodnie z oczekiwaniami czas ten wzrasta proporcjonalnie do liczby incydentów. Wyszukiwanie incydentów trwa dłużej niż dodawanie incydentów, ale w tym przypadku średnie czasy są również akceptowalne dla użytkownika końcowego, ponieważ nie przekraczają nawet 170 ms.



Rysunek 8.3: Czas wyszukiwania incydentów w zależności od liczby incydentów w bazie danych

Ocena testów wydajnościowych

Przeprowadzone testy wydajnościowe wykazały, że najważniejsze funkcje systemu: dodawania nowych incydentów oraz wyszukiwania incydentów, wykonywane są w akceptowalnym czasie.

Założymy, że firma posiada tysiąc klientów. Każdy klient zgłasza jeden incydent na miesiąc. Jeżeli czas działania systemu wynosi cztery lata, to system powinien obsłużyć 48 tys. incydentów. Przeprowadzone testy udowodniły również, że system potrafi obsłużyć taki wolumen danych.

Rozdział 9

Podsumowanie

ITIL jest to zbiór praktyk umożliwiający: redukcję kosztów poprzez zarządzanie IT w sposób analogiczny do zarządzania biznesem, skupienie uwagi na potrzebach klientów oraz ciągłe doskonalenie organizacji. Autor zrealizował fragment systemu Service Desk, zgodny z zaleceniami ITIL v3, obejmujący procesy: zarządzania incydentami, zarządzania problemami i zarządzania zdarzeniami. Zrealizowany fragment systemu może zostać wykorzystany przez małej lub średniej wielkości firmę w dziale obsługi klienta. Proces zarządzania incydentami umożliwia zwiększenie dostępności oferowanych usług IT oraz dostosowanie działalności IT do priorytetów biznesowych. Natomiast proces zarządzania problemami umożliwia szybsze rozwiązywanie incydentów oraz identyfikację trwałych rozwiązań, co wpływa na zredukowanie liczby incydentów.

Zrealizowany system jest łatwo konfigurowalny. Każda organizacja może dostosować wartości słowników do własnych potrzeb. Wszystkie komunikaty zawarte są w pliku właściwości, co pozwala na modyfikację dowolnego komunikatu bez ingerencji w kod aplikacji, dodatkowo umożliwia to zmianę języka. Największą zaletą systemu jest elastyczne rozwiązywanie dla eskalacji: sygnały. System pozwala na utworzenie dowolnej liczby poziomów eskalacji, określenie odbiorców komunikatów, konfigurację treści komunikatów oraz kryteriów eskalacji. Co więcej, zaproponowane rozwiązanie może zostać wykorzystane do przypominania pracownikom o dowolnych zadaniach do zrealizowania.

Istniejący fragment systemu można zintegrować z fragmentem systemu realizowanym przez Adriana Wiśniewskiego w ramach dyplomowej pracy inżynierskiej obejmującym procesy: zarządzania konfiguracją oraz zarządzania zmianą. Proces zarządzania konfiguracją umożliwia lepszą analizę przyczyny powstawania danego incydentu. Natomiast proces zarządzania zmianą pozwoli na kontrolowanie cyklu życia zmian. Jest to istotny proces, ponieważ rozwiązywanie problemu często wymaga wprowadzenia zmiany w infrastrukturze IT. Fragment zrealizo-

wany w ramach niniejszej pracy we współpracy z fragmentem realizowanym przez Adriana Wiśniewskiego obejmuje wszystkie procesy Service Support z ITIL v2.

Przy implementacji systemu wykorzystano nowoczesne narzędzia wspierające prace programisty, projektanta oraz analityka. Skorzystano z nowoczesnego środowiska IDE SpringSource Tool Suite, które wspiera tworzenie aplikacji WWW w technologii J2EE opartych o Spring Framework oraz Hibernate. Użyto narzędzia JDeveloper, które umożliwiło utworzenie diagramów tabel, z których został wygenerowany skrypt SQL. Wykorzystano również narzędzie Rational Software Architect do utworzenia wstępniego projektu: diagramów klas, diagramów sekwencji i diagramów aktywności. Do implementacji raportów wykorzystano gotowe rozwiązanie Open Source: Jasper Reports. Utworzona aplikacja, jak na fragment systemu, jest stosunkowo złożona. Kod źródłowy zawiera 10 KLOC¹ (bez komentarzy oraz pustych linii), testy jednostkowe 2 KLOC, konfiguracyjne pliki XML 1,1 KLOC, strony JSP 2,3 KLOC. Baza danych składa się z 56 tabel.

Dalszy kierunek rozwoju systemu to zwiększenie poziomu automatyzacji. Nowoczesne systemy Service Desk dokonują analizy CMDB przy określaniu wpływu incydentu. Kolejnym nieroziązonym problemem pozostaje naniesienie kalendarza przy liczeniu czasu rozwiązania incydentów. System powinien uwzględniać dni robocze, godziny pracy serwisantów oraz godziny dostępności usługi. Niektóre systemy Service Desk umożliwiają naniesienie wielu kalendarzy, jest to potrzebne, gdy klienci mają różne godziny dostępności dla tej samej usługi. Kolejny aspekt związany z automatyzacją i kalendarzem to przydzielanie zgłoszeń serwisantom. Nowoczesny system powinien przydzielać zgłoszenia serwisantom w sposób sprawiedliwy uwzględniając obciążenie, dni robocze i urlopy. Nie rozwiązano również problemu kontroli widoczności incydentów dla poszczególnych ról. Przykładowo serwisant widzi w systemie tylko swoje incydenty, menedżer grupy ma dostępne incydenty przypisane do jego grupy. Oczywiście nie każda organizacja chce narzucać takie poziomy widoczności. Wynika z tego, że taka kontrola powinna być łatwo konfigurowalna, aby było możliwe utworzenie wielu poziomów widoczności. Przy realizacji tego wymagania warto rozważyć wykorzystanie Oracle Virtual Private Database. Następny problem związany jest z raportowaniem. Dla systemu Service Desk powinien zostać utworzony Data Mart, który będzie umożliwiał tworzenie wielowymiarowych analiz. Aktualne istniejące raporty mają charakter operacyjny. Podejście to wymaga tworzenia zbyt wielu raportów. Istnieją systemy zawierające ponad sto różnych raportów. Nie jest to najlepsze podejście, ponieważ nikt nie dysponuje wystarczającą ilością czasu do przeglądania takiego ogromu informacji. Użytkownicy powinni mieć możliwość tworzyć raporty osobście w sposób analogiczny jak są tworzone tabele przestawne w programie Excel. Potrzebne jest również umożliwienie poruszania po hierarchii wymiarów za pomocą opcji Drill Up, Drill Down. Przy realizacji tego wymagania warto rozważyć

¹KLOC - Kilo Lines Of Code

narzędzie Business Objects lub podobne.

Bibliografia

- [1] Ben Alex and Luke Taylor. Spring security - reference documentation. <http://static.springsource.org/spring-security/site/docs/3.0.x/reference/springsecurity.html>. [cytowanie na str. 65]
- [2] Joel Birch. jquery superfish. http://users.tpg.com.au/j_birch/plugins/superfish/. [cytowanie na str. 66]
- [3] Alison Cartlidge, Ashley Hanna, Colin Rudd, Ivor Macfarlane, John Windebank, and Stuart Rance. *An Introductory Overview of ITIL® V3*. The UK Chapter of the itSMF, 2007. [cytowanie na str. 3]
- [4] Brian Ghidinelli. jquery table sorter - reference documentation. <http://tablesorter.com/docs/>. [cytowanie na str. 66]
- [5] David Heffelfinger. *JasperReports for Java Developers: Create, Design, Format and Export Reports with the world's most popular Java reporting library*. Packt Publishing, 2006. [cytowanie na str. 66]
- [6] Portal ITLife.pl. Korzenie itil. <http://itsm.itlife.pl/content/view/10012/57/>. [cytowanie na str. i, 6, 7, 10]
- [7] Jaspersoft. ireport. <http://jasperforge.org/projects/ireport/>. [cytowanie na str. 65]
- [8] Rod Johnson, Juergen, and Hoeller Keith Donald. Spring framework - reference documentation. <http://static.springsource.org/spring-ws/sites/1.5/reference/html/index.html>. [cytowanie na str. 65]
- [9] Kiczales, Lamping, Mendhekar, Maeda, Lopes, and Loingtier. Aspect-oriented programming. *Irwin-Proceedings European Conference on Object-Oriented Programming*, 1997. [cytowanie na str. 59]
- [10] Christian Bauer Gavin King. *Hibernate w akcji*. Helion, 2007. [cytowanie na str. 65]
- [11] Gavin King, Christian Bauer, Max Rydahl Andersen, Emmanuel Bernard, and Steve Ebersole. Hibernate reference documentation. <http://docs.jboss.org/hibernate/stable/core/reference/en/html/>. [cytowanie na str. 65]
- [12] Cody Lindley. *jQuery Cookbook*. O'Reilly Media, 2009. [cytowanie na str. 66]

- [13] Gary Mak. *Spring Recipes: A Problem-Solution Approach.* Apress, 2008.
[cytowanie na str. 59, 90]
- [14] Office of Gevernment Commerce. *Continual improvmment.* The Stationery Office, 2007. [cytowanie na str. i, 13]
- [15] Office of Gevernment Commerce. *Service design.* The Stationery Office, 2007.
[cytowanie na str. i, 10]
- [16] Office of Gevernment Commerce. *Service operation.* The Stationery Office, 2007.
[cytowanie na str. i, 6, 12, 14, 18, 21, 23]
- [17] Office of Gevernment Commerce. *Service strategy.* The Stationery Office, 2007.
[cytowanie na str. i, 10]
- [18] Office of Gevernment Commerce. *Service transition.* The Stationery Office, 2007.
[cytowanie na str. i, 11]
- [19] Arjen Poutsma, Rick Evans, and Tareq Abed Rabbo. Spring web services - reference documentation. <http://static.springsource.org/spring-ws/sites/1.5/reference/html/index.html>. [cytowanie na str. 65]
- [20] International Data Group Poland S.A. Zarządzanie itsm. Technical report, Computerworld, 2007. [cytowanie na str. 3, 5]
- [21] Richard D. Worth, Scott González, and Todd Parker. jquery ui - documentation. <http://jqueryui.com/demos/>. [cytowanie na str. 66]
- [22] Jorn Zaefferer. jquery treeview - documentation. <http://docs.jquery.com/Plugins/Treeview>. [cytowanie na str. 66]

Dodatek A

Zawartość płyty CD

- kod źródłowy aplikacji,
- niniejszy tekst w formacie PDF,
- dokumentacja techniczna aplikacji wygenerowana przy pomocy JavaDoc,
- projekt JDeveloper zawierający diagramy tabel,
- skrypt SQL tworzący bazę danych: utworzenie tabel oraz wypełnienie tabel przykładowymi danymi,
- wstępny projekt utworzony w Rational Software Architect,
- aplikacja w formacie jar.

Dodatek B

Instalacja

B.1 Pliki konfiguracyjne

B.1.1 `jdbc.properties`

Plik konfiguracyjny definiujący połączenie z bazą danych. Zawiera następujące parametry:

`jdbc.driverClassName` – nazwa klasy do połączenia z bazą danych,
`jdbc.url` – adres url bazy danych,
`jdbc.username` – nazwa użytkownika w bazie danych,
`jdbc.password` – hasło użytkownika w bazie danych.

B.1.2 `messages.properties`

Plik zawierający treści komunikatów wyświetlanego w aplikacji. Daje to możliwość zmiany w dowolnym momencie treści dowolnego komunikatu, bez ponownej komplikacji aplikacji. Każdy komunikat jest postaci klucz, wartość.

B.2 Skrypty SQL

B.2.1 `ddl.sql`

Skrypt SQL tworzący wszystkie obiekty w bazie danych: tabele, perspektywy, wyzwalacze, pakiety, procedury składowane, indeksy.

B.2.2 `inserts.sql`

Skrypt SQL wypełniający tabele przykładowymi danymi.

B.3 Zainstalowanie aplikacji

B.3.1 Zainstalowanie aplikacji na serwerze Tomcat przy pomocy Tomcat Manager

Po uruchomieniu aplikacji Tomcat Manager, w zakładce Deploy należy wybrać plik **ServiceDesk.jar**, następnie dokonać potwierdzenia poprzez kliknięcie w przycisk Deploy.

Spis symboli i skrótów

Skrót	Opis
AOP	Aspect Oriented Programming
CAB	Change Advisory Board
CAPEX	Capital Expenditures
CI	Configuration Item
CMDB	Configuration Management Database
CMMI	Capability Maturity Model Integration
COBIT	Control Objectives for Information and related Technology
DI	Dependency Injection
GITIMM	Government Information Technology Infrastructure Management Method
IoC	Inversion of Control
ITIL	Information Technology Infrastructure Library
ITSM	Information Technology Service Management
JPA	Java Persistence Annotations
KEDB	Known Error Database
KLOC	Kilo Lines of Code
OGC	Office of Government Commerce
OLA	Operational Level Agreement
OPEX	Operating Expenditures
RAD	Rapid Application Development
RFC	Request For Change
SLA	Service Level Agreement
TDD	Test Driven Development
UC	Underpinning Contract
WSDL	Web Services Description Language

Spis rysunków

2.1	Procesy ITIL2 www.actis-ingenieure.com/ITIL.jpg	7
2.2	Obszary ITIL3 http://itsm.itlife.pl/images/itsm/lifecycle4.jpg	9
2.3	Zarządzanie incydentami - proces	15
2.4	Zarządzanie problemami - proces	19
2.5	Zarządzanie zdarzeniami - proces	21
3.1	Katalog usług firmy Red Host S.A.	26
5.1	Diagram tabel dla zarządzania incydentami	45
5.2	Diagram tabel dla zarządzania problemami	47
5.3	Diagram tabel dla zarządzania zdarzeniami	49
5.4	Diagram tabel dla zarządzania zleceniami	50
5.5	Diagram tabel dla sygnałów	52
5.6	Diagram tabel dla zarządzania pracownikami	53
5.7	Diagram tabel dla zarządzania katalogiem usług	54
5.8	Diagram tabel dla zarządzania poziomem usług	55
5.9	Diagram sekwencji dotyczący edycji incydentu	57
6.1	Diagram pakietów dla zarządzania incydentami	68
6.2	Diagram pakietów dla zarządzania problemami	69
6.3	Diagram pakietów dla zarządzania zdarzeniami	71
6.4	Diagram pakietów dla zarządzania zleceniami	72
6.5	Diagram pakietów dla zarządzania sygnałami	74
7.1	Formularz nowego incydentu	77
7.2	Formularz nowego incydentu - szczegóły	78
7.3	Karta incydentu - przypisanie serwisanta	78
7.4	Karta incydentu - dane podstawowe	79

7.5 Karta incydentu - szczegóły	79
7.6 Sygnały na stronie głównej	80
7.7 Karta incydentu - historia	80
7.8 Karta incydentu - dodanie problemu	81
7.9 Karta problemu - powiązane incydenty	81
7.10 Karta problemu - dane podstawowe	81
7.11 Karta problemu - dane szczegóły	82
7.12 Karta problemu - analiza	82
7.13 Karta problemu - analiza	83
7.14 Karta problemu - znany błąd	83
7.15 Wyszukiwanie znanych błędów	83
7.16 Karta incydentu - rozwiążanie	84
7.17 Karta incydentu - szczegóły zgłaszającego	84
8.1 Pokrycie testów jednostkowych	87
8.2 Czas dodania incydentu w zależności od liczby incydentów w bazie danych	91
8.3 Czas wyszukiwania incydentów w zależności od liczby incydentów w bazie danych	92

Spis tabel

2.1	Kodowanie priorytetu	17
2.2	Czas rozwiązania	17
5.1	Treść wiadomości na podstawie szablonu	60