|  |  |
| --- | --- |
| Politechnika Warszawska  Wydział Elektroniki i Technik Informacyjnych  Instytut Informatyki | Rok akademicki 2010/2011 |

Praca Dyplomowa Inżynierska

Adrian Wiśniewski

System Service Desk zgodny z zaleceniami ITIL:

Zarządzanie konfiguracją i zmianą

Opiekun pracy:

dr inż. Michał Rudowski

|  |  |
| --- | --- |
| Ocena |  |
| Podpis przewodniczącego  Komisji Egzaminu Dyplomowego |  |

|  |  |
| --- | --- |
|  | Kierunek: Informatyka  Specjalność: Inżynieria Systemów Informatycznych  Data urodzenia: 24 grudnia 1987 r.  Data rozpoczęcia studiów: październik 2006 r. |

Życiorys

Urodziłem się 24 grudnia 1987 r. w Nowym Dworze Mazowieckim. Mieszkam w Wołominie, gdzie ukończyłem Sportową Szkołę Podstawową nr 5 im. C. K. Norwida i Sportowe Gimnazjum nr 3 im. Polskich Olimpijczyków. Następnie uczęszczałem do klasy o profilu matematyczno-fizycznym w IV Liceum Ogólnokształcącym im. Adama Mickiewicza w Warszawie. W październiku 2006 r. rozpocząłem studia na Politechnice Warszawskiej na Wydziale Elektroniki i Technik Informacyjnych na kierunku Informatyka.

Pierwszą pracę jako programista aplikacji internetowych podjąłem rok później —w 2007 r. — w firmie Psiloc Sp. z o. o., gdzie rozwijałem aplikację www.locatik.pl. Następnie pracowałem na zlecenie przy wielu różnych projektach dla klientów takich jak McDonald’s, Allegro, czy gazeta.pl.

Ponadto od 2006 roku należę do społeczności twórców gier komputerowych www.gamedev.pl, a w latach 2009 – 2010 należałem do Koła Naukowego Twórców Gier Komputerowych „Polygon”, gdzie pełniłem rolę administratora i członka zarządu koła.

|  |  |
| --- | --- |
|  | Podpis studenta |

Egzamin dyplomowy

Złożył egzamin dyplomowy dn. 2011 r.

Z wynikiem

Ogólny wynik studiów

Dodatkowe wnioski i uwagi Komisji

Streszczenie

Tutaj streszczenie

Abstract

Tutaj streszczenie ang

Spis treści

[Rozdział 1. Wstęp 9](#_Toc298844556)

[1.1. Cel pracy 10](#_Toc298844557)

[1.2. Motywacja 10](#_Toc298844558)

[1.3. Zakres 11](#_Toc298844559)

[1.4. Układ 11](#_Toc298844560)

[1.5. Nazewnictwo 11](#_Toc298844561)

[Rozdział 2. ITSM 13](#_Toc298844562)

[2.1. Usługa 13](#_Toc298844563)

[2.2. Proces 13](#_Toc298844564)

[Rozdział 3. ITIL 15](#_Toc298844565)

[3.1. Historia 16](#_Toc298844566)

[3.1.1. Wersja pierwsza 16](#_Toc298844567)

[3.1.2. Wersja druga 17](#_Toc298844568)

[3.1.3. Wersja trzecia 18](#_Toc298844569)

[3.2. Struktura 18](#_Toc298844570)

[3.2.1. Strategia usług (Service Strategy) 20](#_Toc298844571)

[3.2.2. Projektowanie usług (Service Design) 20](#_Toc298844572)

[3.2.3. Przekazanie usług (Service Transition) 20](#_Toc298844573)

[3.2.4. Eksploatacja usług (Service Operation) 21](#_Toc298844574)

[3.2.5. Ustawiczne doskonalenie usług (Continual Service Improvement) 21](#_Toc298844575)

[3.3. Wyszczególnione procesy 21](#_Toc298844576)

[3.3.1. Zarządzanie konfiguracją 21](#_Toc298844577)

[3.3.2. Zarządzanie zmianą 21](#_Toc298844578)

[3.4. Inne metodyki 21](#_Toc298844579)

[Rozdział 4. Firma RedHost 23](#_Toc298844580)

[4.1. Charakterystyka 23](#_Toc298844581)

[4.1.1. Świadczone usługi 23](#_Toc298844582)

[4.2. Wprowadzenie zarządzania konfiguracją 23](#_Toc298844583)

[4.2.1. Przykładowe elementy konfiguracji 23](#_Toc298844584)

[4.3. Wprowadzenie zarządzania zmianą 23](#_Toc298844585)

[4.3.1. Przykładowe żądania zmian 23](#_Toc298844586)

[Rozdział 5. Wymagania 25](#_Toc298844587)

[5.1. Technologia (TECH) 25](#_Toc298844588)

[5.2. Architektura (ARCH) 26](#_Toc298844589)

[5.3. Moduł zarządzania konfiguracją (KONF) 26](#_Toc298844590)

[5.4. Moduł zarządzania zmianą (ZMIA) 29](#_Toc298844591)

[5.5. Rdzeń systemu (CORE) 31](#_Toc298844592)

[Rozdział 6. Projekt 33](#_Toc298844593)

[Rozdział 7. Implementacja 35](#_Toc298844594)

[Rozdział 8. Użytkowanie systemu 37](#_Toc298844595)

[Rozdział 9. Testy i ocena wydajności 39](#_Toc298844596)

[Rozdział 10. Podsumowanie 41](#_Toc298844597)

[Bibliografia 43](#_Toc298844598)

[Dodatek A. Zawartość płyty CD 45](#_Toc298844599)

[Wykaz skrótów 47](#_Toc298844600)

[Spis Tabel 49](#_Toc298844601)

[Spis ilustracji 51](#_Toc298844602)

# Wstęp

W dobie społeczeństwa informacyjnego komputery stały się nieodłącznym elementem naszego życia. Są dziś atrybutem nowoczesności i motorem napędowym postępu, a ich ogromny wpływ na zmiany w gospodarce jest niekwestionowany. Przedsiębiorcy od największych korporacji do małych firm korzystają z informatyki, która jest już nierozerwalnym elementem biznesu. Systemów tworzonych na potrzeby przedsiębiorczości przybywa każdego dnia i są one coraz bardziej skomplikowane biorąc pod uwagę zarówno złożoność rozwiązywanych problemów, ilość wykorzystywanych zasobów, jak i integrację z innymi systemami.

Dynamiczny rozwój infrastruktury informatycznej przedsiębiorstw stawia nowe wyzwania w dziedzinie zarządzania. Dotychczas zadania działu IT skupiały się na technologii i dostarczaniu pojedynczych elementów takich jak aplikacja, serwer, sieć, czy baza danych. Postępowanie takie skutkuje rozproszeniem infrastruktury w przedsiębiorstwie, co z kolei znacznie utrudnia zarządzanie. Brak jednolitych procedur uniemożliwia efektywne przeprowadzanie audytów oraz skutkuje obniżeniem jakości świadczonych usług, co kończy się awariami. Ponadto zasoby systemów są od siebie odizolowane i niemożliwy jest ich transfer, a w konsekwencji dopasowanie do zmiennego w czasie zapotrzebowania użytkowników. Systemy przez większość czasu nie korzystają z pełni dostępnej mocy, co jest marnotrawstwem i może być postrzegane jako strata dla przedsiębiorstwa.

Model ITSM (ang. IT service management) podchodzi do problemu w zupełnie odwrotny sposób. Traktuje on infrastrukturę IT jako całość i skupia się na postrzeganiu jej z perspektywy użytkownika. Nie jest ważne jak i z czego zbudowane są poszczególne systemy, natomiast kluczowe znaczenie ma fakt, że świadczą one usługi, istotne dla osiągnięcia celów biznesowych przedsiębiorstwa. Usługą IT nazywamy zbiór sprzętu komputerowego, oprogramowania, łącz oraz innych elementów infrastruktury; które razem zapewniają spójną funkcjonalność pozwalającą uzyskać zamierzony efekt. Warto zwrócić uwagę, że definicja ta ma charakter uniwersalny. Pozwala to na grupowanie typowych czynności związanych z zarządzaniem usługami IT w procesy, które dzięki swojej ogólności mogą być stosowane niezależnie od specyfiki przedsiębiorstwa. Ponadto funkcjonowanie działu IT w roli dostawcy usług pozwala skupić się bezpośrednio na celach biznesowych, ponoszonych kosztach oraz wymaganej jakości; podczas gdy kwestie techniczne są rozstrzygane wewnątrz departamentu informatycznego przez odpowiednio wykwalifikowany personel.

ITSM jedynie definiuje sposób w jaki należy postrzegać infrastrukturę informatyczną. Nie jest gotową receptą na problemy zarządzania IT, lecz jako nowatorskie podejście stanowi punkt wyjścia dla wielu konkretnych metodyk. Bez wątpienia obecnie najbardziej popularną i najszerzej stosowaną z nich jest ITIL (ang. Information Technology Infrastructure Library). Jest to kodeks postępowania opracowany przez OGC[[1]](#footnote-1), znane z wprowadzenia innych modeli zarządzania takich jak PRINCE2, MSP, czy M\_o\_R. ITIL składa się ze zbioru najlepszych praktyk, wypróbowanych w wielu przedsiębiorstwach, co zarazem stanowi jego główną zaletę. Nie narzuca sztywnych wytycznych, przez co może być przystosowany do różnych potrzeb. Ponadto — jako metodyka wytworzona przez organizacje rządowe — nie jest własnością żadnej firmy, dzięki czemu jest niezależna od platformy technologicznej i specyfiki branży.

## Cel pracy

Stworzenie prototypu systemu do zarządzania infrastrukturą informatyczną przedsiębiorstwa, opartego o ITIL w wersji 3. Ponadto zobrazowanie zastosowań tego systemu na podstawie przypadków użycia w przykładowej firmie.

## Motywacja

Na rynku istnieją gotowe, skomplikowane i duże produkty do zarządzania usługami IT oparte o ITIL, a ogromne koszty licencji sprawiają, że są one dostępne jedynie dla największych firm. Dodatkowo często występującym problemem jest brak możliwości adaptacji systemu do zmieniającej się charakterystyki przedsiębiorstwa. Jest to spowodowane przechowywaniem informacji na temat elementów konfiguracji w relacyjnych bazach danych w tradycyjny sposób, gdzie informacje o elementach jednego typu znajdują się w jednej tabeli. Schemat bazy danych tworzony jest na sztywno, a w konsekwencji dodanie nowych atrybutów lub typów elementów wymaga zarówno zmian w bazie danych jak i modyfikacji systemu. Moja praca dyplomowa jest próbą stworzenia niedrogiego, intuicyjnego i adaptowalnego narzędzia, które będzie mogło być stosowane także przez średnie i małe przedsiębiorstwa.

## Zakres

ITIL jest standardem złożonym z bardzo dużej liczby procesów. Wykonanie systemu wspierającego wszystkie byłoby zadaniem przerastającym siły niejednej firmy, nie wspominając o możliwościach dyplomanta. W związku z tym z całego standardu zostały wybrane najważniejsze procesy związane z przekazaniem usług i ich eksploatacją.

Następnie procesy z obu tych grup zostały równolegle zaimplementowane w ramach dwóch prac dyplomowych inżynierskich. Procesy z grupy przekazania usług — zarządzanie konfiguracją i zmianą — są opisane przez autora w niniejszej pracy. Pozostałe procesy z drugiej grupy — zarządzanie incydentami i problemami — zostały opisane przez Piotra Kalańskiego[[2]](#footnote-2).

Należy zaznaczyć, że w ramach współpracy autorów obu prac powstał jeden system, w którym wszystkie zaimplementowane procesy współdziałają ze sobą. Prace związane z projektowaniem i implementacją współużytkowanych części systemu oraz opisaniem przykładowej firmy zostały wykonane wspólnie.

## Układ

Rozdział drugi opisuje zbiór zaleceń ITIL z szczególnym uwzględnieniem zarządzani konfiguracją i zmianą. W rozdziale trzecim przedstawiono profil przykładowej firmy, która zamierza wdrożyć ITIL wraz z systemem informatycznym automatyzującym jego procesy. Rozdziały czwarty, piąty i szósty omawiają kolejne etapy wykonania tego systemu, odpowiednio: zebranie wymagań funkcjonalnych i niefunkcjonalnych, projekt oraz implementację. Następujący rozdział siódmy prezentuje zastosowania systemu po jego wdrożeniu w opisanej wcześniej firmie. Rozdział ósmy skupia się na omówieniu testów poprawności i wydajności implementacji. Kończący pracę rozdział dziewiąty stanowi domknięcie i podsumowanie całości.

## Nazewnictwo

Wszystkie nazwy związane z ITIL występujące w tej pracy zostały przetłumaczone zgodnie z oficjalnym glosariuszem wydanym przez OGC[[3]](#footnote-3).

# ITSM

## Usługa

## Proces

# ITIL

2.5 + 1.5 + 6 + 7 = 17

## Historia

Prace nad ITIL zaczęły się ponad trzydzieści lat temu — w drugiej połowie lat osiemdziesiątych. Rząd Wielkiej Brytanii szukał sposobów na ograniczenie wydatków związanych z utrzymaniem systemów informatycznych, które sięgały kwoty około 8 miliardów funtów w skali roku[[4]](#footnote-4). Źródła wysokich kosztów dopatrywano się przede wszystkim w fakcie, że wszystkie departamenty i instytucje rządowe zarządzały infrastrukturą IT we własnym zakresie. Nie istniały jednolite procedury i standardy, a co za tym idzie możliwości przeprowadzania audytów i kontroli wydatków były znacznie ograniczone. W celu rozwiązania problemu zlecono komórce rządowej o nazwie Central Computer and Telecommunications Agency (CCTA) opracowanie metody zarządzania pozwalającej na efektywne wykorzystanie zasobów IT.

Tak w roku 1988[[5]](#footnote-5) powstał GITIMM (ang. Government Information Technology Infrastructure Management Method) będący prekursorem ITIL. Myślą przewodnią nowej metodyki było ujednolicenie sposobu zarządzania IT. Dzięki wprowadzeniu wspólnego języka i stosowaniu podobnych praktyk możliwa stała się profesjonalizacja pracowników zarządzających infrastrukturą IT, co w konsekwencji pozwoliło podnieść efektywność i obniżyć koszty. Ponadto metodyka zawierała bardzo wiele cennych i sprawdzonych porad opartych na doświadczeniach wielu różnych firm, z którymi prowadzono konsultacje. W szczególności skorzystano z opracowanych przez IBM i wydanych w 1980 r. publikacji z serii „A Management System for the Information Business” opisujących koncepcję ITSM. GITIMM spełnił oczekiwania i został szybko przyjęty przez sektor prywatny oraz instytucje rządowe nie tylko Wielkiej Brytanii.

### Wersja pierwsza

Rok później, w sierpniu 1989, podczas krótkiej rozmowy dwóch twórców zmieniono nazwę na ITIL[[6]](#footnote-6). Zmiana była kluczowa, ponieważ z nazwy zniknęło słowo „government” oznaczające rząd, które odstraszało wielu przedsiębiorców. Ponadto zrezygnowano ze słowa „metoda”, ponieważ metodyka ta jest w rzeczywistości zbiorem wskazówek, a nie sztywnych wytycznych. W ciągu kilku lat wydano łącznie 42 tomy składające się na pierwszą wersję ITIL[[7]](#footnote-7). Do popularyzacji i dalszego rozwoju metodyki przyczyniła się utworzona w 1991 r. [[8]](#footnote-8) w internecie grupa użytkowników ITIMF (ang. IT Infrastructure Management Forum), obecnie funkcjonująca pod nazwą itSMF (ang. IT Service Management Forum) jako organizacja non-profit szerząca wiedzę o ITSM.

### Wersja druga

W 2000 roku podjęto próbę uporządkowania i ujednolicenia metodyki. Efektem był podział ITIL na dwa kluczowe obszary zawierające łącznie jedenaście dyscyplin. Pierwszy obszar — wspieranie usług (ang. Service Support) — obejmuje sześć dyscyplin związanych z technologią, niezbędnych do zapewnienia wysokiej jakości świadczonych usług:

|  |  |
| --- | --- |
| Nazwa dyscypliny | Opis |
| Service Desk | Zapewnia pojedynczy punktu kontaktu dla użytkowników końcowych potrzebujących wsparcia. Skupia się na komunikacji z klientem i kontroli incydentów. Jedyna dyscyplina będąca funkcją a nie procesem. |
| Zarządzanie incydentami  (Incident Management) | Głównym celem tej dyscypliny jest przywrócenie działania usługi po wystąpieniu incydentu w jak najkrótszym czasie i przy uwzględnieniu obowiązujących SLA[[9]](#footnote-9). Incydent to zdarzenie, które wpływa lub może wpływać na obniżenie jakości świadczonej usługi. |
| Zarządzanie problemami  (Problem Management) | Skupia się na znalezieniu problemów będących źródłem incydentów i podjęciu działań korygujących. W trakcie naprawy udostępnia informacje o znanych błędach i metodach ich obejścia. |
| Zarządzanie zmianami  (Change Management) | Zapewnia standardowe procedury zarządzania wszelkimi zmianami dotyczącymi usług. Minimalizuje ich negatywny wpływ poprzez odpowiednią ocenę, autoryzację i planowanie każdej zmiany. |
| Zarządzanie wersjami  (Release Management) | Grupuje zmiany w wersje i odpowiada za skoordynowanie ich implementacji oraz wdrożenia. |
| Zarządzanie konfiguracją  (Configuration Management) | Zajmuje się przechowywaniem i ciągłą aktualizacją informacji o wszystkich komponentach potrzebnych do świadczenia usług. Stanowi bazę wiedzy, z której korzystają inne procesy. |

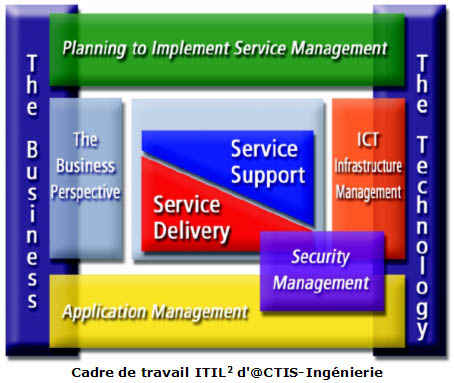
Tabela .. Dyscypliny wspierania usług ITIL v2

Drugi obszar — świadczenie usług (ang. Service Delivery) — jest bliższy perspektywie biznesowej i dotyczy przestrzegania gwarancji danych klientowi przez świadczącego usługi. Składa się z pięciu dyscyplin:

|  |  |
| --- | --- |
| Nazwa dyscypliny | Opis |
| Zarządzanie poziomem usług  (Service Level Management) | Pozwala na dopasowanie usług IT do wymagań biznesu. Skupia się na określeniu przejrzystych oczekiwań i wymagań klienta w stosunku do pobieranych przez niego usług. Dyscyplina ta zajmuje się tworzeniem katalogu usług, zbieraniem wymagań i negocjacją SLA. |
| Zarządzanie finansami  (Financial Management) | Zajmuje się obliczanie kosztów świadczonych usług ich księgowaniem oraz ustalaniem budżetu. Pozwala na funkcjonowanie wewnętrznych departamentów IT jako jednostek biznesowych. |
| Zarządzanie dostępnością  (Availability Management) | Celem tej dyscypliny jest zapewnienie dostępności usług zgodnie z SLA w efektywny kosztowo sposób. Wymaga to analizy, planowania, mierzenia i ewentualnej poprawy wszystkich czynników mających wpływ na dostępność. |
| Zarządzanie pojemnością  (Capacity Management) | Dyscyplina podobna do zarządzania dostępnością, zorientowana na monitorowanie i przestrzeganie gwarancji dotyczących pojemności świadczonych usług. |
| Zarządzanie ciągłością  usług informatycznych  (IT Service Continuity Management) | Znana także jako zarządzanie sytuacją kryzysową. Głównym celem jest przywrócenie działania systemów informatycznych firmy w uzgodnionym czasie. Składa się z zapobiegania wystąpieniu sytuacji kryzysowych i opracowaniu planów przeciwdziałania im. |

Tabela .. Dyscypliny świadczenia usług ITIL v2

Po wydaniu pierwszych dwóch tomów dotyczących wsparcia i świadczenia usług powstało jeszcze pięć następnych opisujących inne obszary i dwa dotyczące odpowiednio ITSM oraz stosowania ITIL w małych przedsiębiorstwach. W tym czasie doszło do restrukturyzacji instytucji rządowych Wielkiej Brytanii, na skutek której w 2001 r. OGC (ang. Office of Government Commerce) wchłonęło CCTA i tym samym stało się posiadaczem praw do ITIL. Druga wersja wprowadziła jasno wydzielone procesy i znacząco uporządkowała czynności opisane w wersji pierwszej. Skonsolidowano wiedzę zawartą w czterdziestu dwóch tomach do zaledwie dziewięciu.



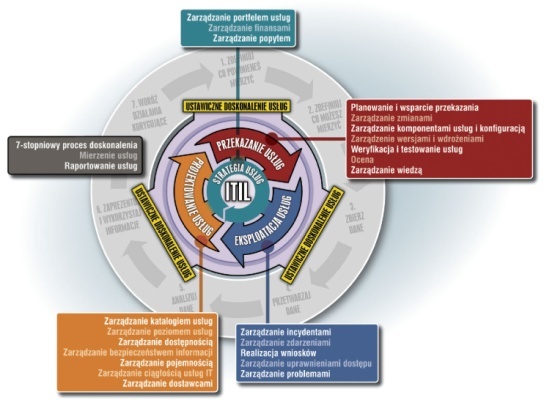
Ilustracja .. Obszary ITIL v2[[10]](#footnote-10)

### Wersja trzecia

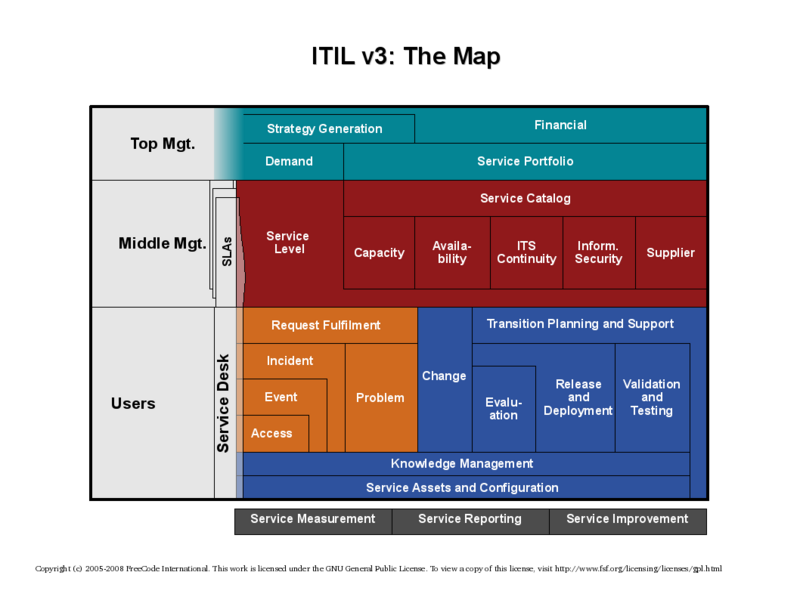
W listopadzie 2004 r.[[11]](#footnote-11) OGC uruchomiło projekt ITIL Refresh, którego celem było stworzenie kolejnej wersji ITIL. Projekt w dużej mierze polegał na przeprowadzaniu konsultacji z tysiącami organizacji, które używały ITIL, i skorzystaniu z ich opinii oraz doświadczeń. W maju 2007 r. projekt został zakończony, a jego rezultatem było opublikowanie pięciu tomów ITIL w wersji trzeciej. Zmian było bardzo dużo, zarówno w strukturze metodyki, poszczególnych procesach, jak i samej filozofii zarządzania infrastrukturą IT. Zakres poszerzono blisko dwukrotnie, zwiększając liczbę procesów i funkcji prawie trzykrotnie. W lipcu 2010 r. zaprzestano prowadzenia egzaminów ITIL v2, a dokładnie rok później zaprzestano publikacji materiałów i oficjalnie wycofano wsparcie dla tej wersji.

## Struktura

Rgergerg dghrthtr



Ilustracja .. Cykl życia usługi w ITIL[[12]](#footnote-12)



Ilustracja .. Procesy ITIL pogrupowane wg hierarchii zarządzania[[13]](#footnote-13)

### Strategia usług (Service Strategy)

1

### Projektowanie usług (Service Design)

1

### Przekazanie usług (Service Transition)

1

### Eksploatacja usług (Service Operation)

1

### Ustawiczne doskonalenie usług (Continual Service Improvement)

1

## Wyszczególnione procesy

### Zarządzanie konfiguracją

3

### Zarządzanie zmianą

4

## Inne metodyki

1

# Firma RedHost

Co i dlaczego opis firmy. 0.5 + 4 = 4.5

## Charakterystyka

0.5 + 0.5 = 1

### Świadczone usługi

0.5

## Wprowadzenie zarządzania konfiguracją

0.5 + 1 = 1.5

### Przykładowe elementy konfiguracji

1

## Wprowadzenie zarządzania zmianą

0.5 + 1 = 1.5

### Przykładowe żądania zmian

1

# Wymagania

Przedstawiony zbiór wymagań systemowych stanowi podstawę do wykonania oprogramowania i określa jego zakres. Wszystkie zebrane wymagania zostały podzielone na grupy. Pierwsze dwie — technologia i architektura — zawierają ogólne wymogi dotyczące wykonania systemu, natomiast następne opisują poszczególne elementy systemu. Zarządzanie danymi należy rozumieć jako możliwość wykonywania operacji CRUD[[14]](#footnote-14).

## Technologia (TECH)

Ograniczenia technologii sprowadzają się do wyboru języka programowania i narzędzi. Celem jest stworzenie niezawodnego systemu wysokiej jakości, łatwego w obsłudze i taniego w utrzymaniu.

Spis wymagań niefunkcjonalnych

1. Zastosowanie sprawdzonych i powszechnie znanych technik programowania, aby zmniejszyć koszt i ułatwić utrzymanie systemu: paradygmatu obiektowego wraz ze wzorcami programowania opisanymi w (Gamma, Helm, Johnson i Vlissides, 2005) i (Fowler, 2005).
2. Umożliwienie dostosowania do specyficznych potrzeb klienta poprzez użycie paradygmatu programowania komponentowego, który pozwala na wymianę nawet najmniejszych elementów systemu.
3. Nacisk na enkapsulację poprzez wyodrębnienie funkcjonalności pojawiających się w różnych warstwach systemu (zarządzania transakcjami, autoryzacji i innych), dzięki wykorzystaniu paradygmatu programowania aspektowego.
4. System musi być napisany w języku o statycznym typowaniu, co pozwoli na wczesne wykrycie wielu błędów już na etapie kompilacji.
5. Ilość błędów podczas pracy systemu musi być minimalizowana, poprzez wykorzystanie testów jednostkowych.
6. System do przechowywania informacji musi wykorzystywać relacyjne bazy danych.

## Architektura (ARCH)

Spis wymagań niefunkcjonalnych

1. Wykonanie systemu w postaci trójwarstwowej aplikacji sieciowej z dostępem zdalnym.
2. Wsparcie wielu różnych warstw prezentacji.
   1. Podstawową warstwę prezentacji będzie przeglądarka www.
   2. Drugą stanowią usługi sieciowe[[15]](#footnote-15).
   3. Musi istnieć możliwość późniejszego zaimplementowania innych warstw prezentacji takich jak np. interfejs okienkowy.
3. System musi być zbudowany modularnie i być rozszerzalny.
   1. Rdzeń systemu złożony z wielu modułów będzie udostępniał usługi nie związane bezpośrednio z funkcjonalnością systemu, lecz wymagane przez inne moduły. Przykład: autoryzacja, dostarczanie danych o pracownikach firmy, wysyłanie komunikatów do użytkowników itp.
   2. Drugą grupę będą stanowić moduły odpowiadające procesom ITIL.
   3. Zaimplementowane zostaną procesy zarządzania konfiguracją i zmianą
      1. Procesy te muszą współpracować z zaimplementowanymi przez Piotra Kalańskiego: zarządzaniem incydentami i zmianą.
   4. Musi istnieć możliwość późniejszej implementacji kolejnych procesów ITIL.
   5. Pożądana jest możliwość rozszerzenia systemu o moduły zapewniające funkcjonalność SCM[[16]](#footnote-16), CRM[[17]](#footnote-17) i ERP[[18]](#footnote-18).

## Moduł zarządzania konfiguracją (KONF)

Zadaniem modułu jest zarządzanie CMDB, czyli przechowywanie danych konfiguracji firmy zgodnie z procesem zarządzania konfiguracją ITIL w wersji 3. Zawartość CMDB stanowią informacje o elementach konfiguracji, ich atrybutach i relacjach między nimi. Elementy konfiguracji dzielą się na dwie zasadnicze grupy: elementy wbudowane i uniwersalne.

Do pierwszej należą elementy konfiguracji zarządzanie przez inne moduły takie jak zgłoszenia incydentów, problemów, żądania zmian, ale także pracownicy i klienci. Elementy te są wyświetlane w module konfiguracji w trybie do odczytu. Za edycję elementów wbudowanych odpowiedzialne są ich macierzyste moduły, co pozwala na egzekwowanie odpowiednich uprawnień i ograniczeń wewnątrz tych modułów.

Drugą grupę stanowią elementy uniwersalne posiadające dowolny zestaw atrybutów, określany przez użytkownika. Ich zadaniem jest przechowywanie informacji o elementach konfiguracji specyficznych dla danej branży. Przykładami są: monitory, serwery, elementy infrastruktury, komponenty niezbędne do funkcjonowania formy.

Spis wymagań funkcjonalnych

1. Przechowywanie, wyszukiwanie i prezentacja danych wszystkich elementów konfiguracji (zarówno wbudowanych jak i uniwersalnych).
   1. Element konfiguracji posiada przypisany identyfikator oraz zbiór wartości atrybutów.
   2. Format identyfikatora jest ustalany przez firmę (np. dla monitora LCD zakupionego w roku 2010 roku: HW/MON/LCD/4/30/2010/8).
   3. Dodatkowo element posiada krótkie podsumowanie złożone z wybranych wartości atrybutów, dzięki któremu znacznie łatwiej odnaleźć szukany element na liście (np. dla tego samego monitora: Samsung SME1920N 19" 1360x768 TFT/TN).
2. Zarządzanie uniwersalnymi elementami konfiguracji.
   1. Uniwersalny element konfiguracji posiada dodatkowo typ. Typ podawany jest podczas tworzenia elementu konfiguracji i nie może być zmieniony.
3. Zarządzanie typami elementów uniwersalnych.
   1. Typ elementu uniwersalnego posiada unikalną nazwę, opis oraz zestaw atrybutów.
   2. Typ może być abstrakcyjny, co oznacza, że nie można stworzyć elementu konfiguracji tego typu.
   3. Typy tworzą hierarchię. Podtypy dziedziczą atrybuty typu nadrzędnego.
   4. Kluczowym wymaganiem jest, aby można było dodawać nowe typy z poziomu interfejsu użytkownika, bez potrzeby modyfikacji systemu.
4. Zarządzanie relacjami między elementami konfiguracji.
   1. Relacja łączy dwa elementy konfiguracji (wbudowane lub uniwersalne). Ponadto tak jak w przypadku uniwersalnych elementów konfiguracji posiada typ oraz zbiór wartości atrybutów.
5. Zarządzanie typami relacji. Wszystkie wymagania dotyczące typów uniwersalnych elementów konfiguracji odnoszą się także do typów relacji. A ponadto klasa relacji:
   1. określa czy relacja jest skierowana, czy nie;
   2. wyznacza podpis relacji w obu kierunkach (np. obsługuje, jest obsługiwany przez);
   3. nakłada ograniczenia na typy elementów obu stron relacji.
6. Zarządzanie atrybutami typów elementów uniwersalnych i relacji.
   1. Atrybut posiada nazwę, opis oraz swój format.
      1. W obecnej wersji jedynym dostępnym formatem atrybutu jest napis. W kolejnych wersjach będą dodawane nowe formaty.
7. Zarządzanie wartościami atrybutów elementów uniwersalnych i relacji.
   1. Wartości atrybutów to pary typu atrybut-wartość. Pierwszy element pary musi być atrybutem należącym do typu elementu lub relacji, natomiast drugi musi posiadać format określony przez dany atrybut.

Spis wymagań niefunkcjonalnych

1. Zarządzać elementami konfiguracji i relacjami oraz wartościami ich atrybutów mogą jedynie uprawnieni menadżerowie konfiguracji
2. Zarządzać typami elementów konfiguracji i relacji oraz ich atrybutami mogą jedynie uprawnieni administratorzy konfiguracji.
3. System musi zapewnić miejsce do przechowywania stu tysięcy elementów konfiguracji. Elementy testowe będą posiadać dwadzieścia wartości atrybutów o średniej długości dwustu bajtów.
4. System musi zapewnić miejsce do przechowywania stu typów elementów uniwersalnych i stu typów relacji. Typy testowe będą posiadać dwadzieścia atrybutów o średniej długości wynoszącej jeden kilobajt, ponadto ich nazwa i opis nie przekroczą dwóch kilobajtów.
5. Przy bazie danych zapełnionej w ilości wskazanej w punktach 3. i 4. system powinien odpowiadać na wszystkie zapytania w czasie poniżej pięciu sekund.

## Moduł zarządzania zmianą (ZMIA)

Moduł służy do przechowywania i przetwarzania żądań zmian (RFC) zgodnie z ITIL. Ocenione i autoryzowane zmiany są podstawą do modyfikacji CMDB przy użyciu modułu konfiguracji.

Spis wymagań funkcjonalnych

1. Przyjmowanie nowych żądań zmian.
   1. Zgłaszane nowe żądania zmian posiadają unikalny identyfikator, krótki tytuł, dokładny opis, wstępny priorytet, wstępną kategorię oraz odnotowane: autora i czas złożenia.
   2. Żądanie stanowi element konfiguracji i może być powiązane różnymi relacjami z innymi elementami.
   3. Ponadto żądanie może posiadać komentarze.
   4. Użytkownik może usuwać swoje nowe żądania zmian.
   5. Menadżer konfiguracji może przejąć nowe żądanie zmiany, zmieniając jego stan na nieautoryzowane żądanie zmiany.
2. Zarządzanie nieautoryzowanymi żądaniami zmian.
   1. Nieautoryzowane żądania zmian mogą być filtrowane (odrzucane) przez kierującego nimi menadżera zmian, po podaniu przyczyny.
   2. Menadżer zmiany może zmienić priorytet i kategorię ustawioną przez autora żądania.
   3. Menadżer określa typ zmiany (np. rutynowa, mała, ważna, duża).
   4. Następnie żądanie jest wysyłane do organu autoryzującego zmiany. Organ może różnić się w zależności od kategorii, priorytetu (wraz z pilnością) i typu.
   5. Organ autoryzujący żądanie autoryzuje bądź odrzuca podając przyczynę.
      1. Przy pozytywnym rozpatrzeniu może opcjonalnie podać proponowany czas implementacji zmiany.
3. Zarządzanie autoryzowanymi zmianami.
   1. Wszystkie autoryzowane zmiany trafiają do powiązanej z ich kategorią rady ds. zmian lub w przypadku zmian pilnych — nadzwyczajnej rady ds. zmian.
   2. Rada może zmiany odrzucić lub zmienić ich priorytet i typ.
   3. Rada wpisuje zmiany do harmonogramu zmian oraz określa osobę odpowiedzialną za budowanie i testowanie zmiany. Stan żądania zmiany zmienia się na implementowane.
4. Zarządzanie implementowanym żądaniami zmianami.
   1. Budowniczy po wykonaniu prac zmienia status żądania na gotowe do wdrożenia.
   2. Menadżer zmian koordynuje wdrożenie i odnotowuje jego wynik w żądaniu zmiany.
   3. W przypadku porażki zmiana jest wycofywana i przekazywana z powrotem do rady ds. zmian.
   4. Natomiast w przypadku sukcesu menadżer zmian dołącza do żądania sprawozdanie z implementacji i zamyka żądanie.
5. Zarządzanie odrzuconymi żądaniami zmian.
   1. Dowolny użytkownik może zgłosić apelację od odrzucenia żądania po podaniu odpowiedniego uzasadnienia.
   2. Apelacja może zostać uznana przez menadżera zmian.
   3. Zmiany, których apelacja została uznana, stają się zmianami nieautoryzowanymi.
6. Wyświetlanie harmonogramu zmian.
7. Zarządzanie poziomami priorytetów zmian.
   1. Priorytet ma unikalną nazwę.
   2. Niektóre poziomy mogą być oznaczone jako pilne. Zmiany o priorytecie oznaczonym jako pilny trafiają do nadzwyczajnej rady ds. zmian.
8. Zarządzanie kategoriami żądań zmian.
   1. Kategoria posiada unikalną nazwę.
   2. Kategorie tworzą strukturę hierarchiczną.
   3. Kategoria może posiadać przypisaną zwykłą i nadzwyczajną radę ds. zmian, do której kierowane są wszystkie autoryzowane zmiany danej kategorii. W przeciwnym przypadku zmiany te są wysyłane do rady związanej z kategorią nadrzędną.
9. Zarządzanie typami zmian.
   1. Typ zmiany posiada unikalną nazwę.
10. Zarządzanie organami autoryzującymi.
    1. Organ autoryzujący posiada unikalną nazwę oraz może składać się z jednego lub więcej użytkowników systemu.
    2. Organ otrzymuje zmiany wymagające autoryzacji określonego typu z określonej kategorii.
    3. Szczególnymi rodzajami organu autoryzującego jest Rada ds. Zmian (CAB) i nadzwyczajna Rada ds. Zmian (CAB/EC).

Spis wymagań niefunkcjonalnych

1. Żądania zmian mogą być składane tylko przez uprawnionych użytkowników.
2. Żądania zmian mogą być zarządzane tylko przez uprawnionych menadżerów zmiany.
3. Poziomy priorytetów, kategorie zmian i zakresy wpływu mogą być zarządzane tylko przez uprawnionych administratorów zmian.
4. System musi przechowywać nie mniej niż sto tysięcy wniosków zmian.
5. Czas odpowiedzi systemu na zapytania przy takim obciążeniu nie powinien przekroczyć trzech sekund.

## Rdzeń systemu (CORE)

Zbiór modułów świadczących szeroki wachlarz usług, nie związany bezpośrednio z procesami ITIL, ale uzupełniający je o niezbędną z punktu widzenia użytkownika funkcjonalność.

Spis wymagań funkcjonalnych

1. Uwierzytelnienie tożsamości i logowanie użytkowników do systemu.
   1. Użytkownicy mogą być pracownikami, klientami lub kooperantami.
   2. Uwierzytelnianie bazuje na nazwie użytkownika i haśle.
   3. Każdy użytkownik posiada swój zestaw uprawnień.
   4. Bez odpowiedniej autoryzacji nie można korzystać z systemu.
   5. Po zakończonej pracy użytkownicy mogą się wylogować.
2. Udostępnienie narzędzi komunikacji między użytkownikami
   1. Wysyłanie ogłoszeń do wszystkich użytkowników.
   2. Wysyłanie wiadomości do wskazanych użytkowników.
   3. Przypisywanie użytkownikom zleceń na wykonanie zadania.
      1. Powiadamianie o przebiegu wykonania tych zleceń.
3. Zapisywanie i przechowywanie dowolnych plików w systemie.
   1. Możliwość dodawania plików do wiadomości w formie załączników.
4. Udostępnianie informacji na temat firmy.
   1. Danych pracowników.
   2. Danych klientów.
   3. Danych kooperantów.

Spis wymagań niefunkcjonalnych

1. System musi zapewnić miejsce do przechowywania danych o co najmniej dziesięciu tysiącach użytkowników.
2. System musi mieć możliwość przechowywania łącznie co najmniej miliona ogłoszeń, wiadomości i zleceń.
3. Ilość przechowywanych plików powinna być ograniczona jedynie ilością wolnego miejsca na dyskach twardych serwera.

# Projekt

# Implementacja

# Użytkowanie systemu

# Testy i ocena wydajności

# Podsumowanie

Bibliografia

(2010, Grudzień). Pobrano Lipiec 18, 2011 z lokalizacji Global Journal of Computer Science and Technology: http://computerresearch.org/stpr/index.php/gjcst/article/download/465/424

*ITIL Service Management - A Brief History of ITIL*. (2011, Lipca 18). Pobrano Lipca 18, 2011 z lokalizacji ITIL Service Management: http://itservicemngmt.blogspot.com/2007/09/brief-history-of-itil.html

Fowler, H. (2005). *Architektura systemów zarządzania przedsiębiorstwem. Wzorce projektowe.* Helion.

Gamma, E., Helm, R., Johnson, R. i Vlissides, J. (2005). *Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku.* Wydawnictwa Naukowo Techniczne.

*IT frameworks wiki - Hasło ITIL*. (brak daty). Pobrano Lipiec 7, 2011 z lokalizacji IT frameworks wiki: http://www.itframeworks.org/wiki/Information\_Technology\_Infrastructure\_Library

*IT Life - Co to jest ITIL?* (brak daty). Pobrano Lipiec 7, 2011 z lokalizacji IT Life: http://itsm.itlife.pl/

*IT Life - Korzenie ITIL.* (brak daty). Pobrano Lipiec 18, 2011 z lokalizacji IT Life: http://itsm.itlife.pl/content/view/10012/57/

Kalański, P. (2010). *System Service Desk zgodny z zaleceniami ITIL.*

Sante, T. i Jeroen, E. (2009, Wrzesień). *TOGAF™ 9 and ITIL® V3 Two Frameworks Whitepaper.*

Zawartość płyty CD

Wykaz skrótów

Spis Tabel

Spis ilustracji

Ilustracja 2.1. Cykl życia usługi w ITIL 16

Ilustracja 2.2 Procesy ITIL wg hierarchii zarządzania (źródło: http://www.itframeworks.org) 16

1. Office of Government Commerce — komórka rządu Wielkiej Brytanii. W momencie stworzenia ITIL działająca pod nazwą Central Computer and Telecommunications Agency (CCTA). [↑](#footnote-ref-1)
2. (Kalański, 2010) [↑](#footnote-ref-2)
3. (ITILv3 Glossary Polish, 2008) [↑](#footnote-ref-3)
4. (Global Journal of Computer Science and Technology, 2010) [↑](#footnote-ref-4)
5. (ITIL Service Management - A Brief History of ITIL, 2011) [↑](#footnote-ref-5)
6. (IT Life - Korzenie ITIL) [↑](#footnote-ref-6)
7. (Sante i Jeroen, 2009) [↑](#footnote-ref-7)
8. (Pink Elephant - IT Service Management 2007). [↑](#footnote-ref-8)
9. ang. Service Level Agreement — umowa określająca warunki świadczenia usługi, w szczególności mierzalne parametry dotyczące wydajności i awaryjności. [↑](#footnote-ref-9)
10. źródło: (actis-ingenierie) [↑](#footnote-ref-10)
11. (Projekt ITIL Refresh) [↑](#footnote-ref-11)
12. źródło: (IT Life - Co to jest ITIL?) [↑](#footnote-ref-12)
13. źródło: (IT frameworks wiki - Hasło ITIL) [↑](#footnote-ref-13)
14. ang. Create, Read, Update, Delete. Operacje pozwalające dodawać, odczytywać, uaktualniać i usuwać dane. [↑](#footnote-ref-14)
15. ang. web services [↑](#footnote-ref-15)
16. ang. Supply chain management — Zarządzanie łańcuchem dostaw [↑](#footnote-ref-16)
17. ang. Customer relationship management — Zarządzanie relacjami z klientami [↑](#footnote-ref-17)
18. ang. Enterprise resource planning — Planowanie zasobów przedsiębiorstwa [↑](#footnote-ref-18)