



Standardy dotyczące wytwarzania oprogramowania w Grupie Energa

Standard Dokumentacji Oprogramowania w Grupie Kapitałowej Energia

Wstęp

Niniejszy dokument powstał w związku z wprowadzaniem Katalogu Standardów IT wynikającego z Kodeksu Architektury GK Energia będącego częścią Polityki IT Grupy Energia.

Celem dokumentowania oprogramowania jest utrwalenie i dostarczanie wiedzy na temat budowanego oprogramowania w celu jego dalszego sprawnego rozwoju, utrzymania jak i planowania rozwoju architektury IT.

Uregulowania zawarte w Standardzie Dokumentacji Oprogramowania w Grupie Kapitałowej Energia obowiązują każdą jednostkę organizacyjną realizującą czynności związane z rozwojem i/lub utrzymaniem oprogramowania.

1. Zakres dokumentowania oprogramowania

Wymaganiem jest aby dokumentacja oprogramowania zawierała minimum:

- ✓ Dokumentację Wymagań Biznesowych,
- ✓ Projekt Funkcjonalny (PF),
- ✓ Projekt Techniczny (PT),
- ✓ Dokumentację Techniczną (DT),
- ✓ Dokumentację Administratora (DA),
- ✓ Dokumentację Użytkownika (DU).

Wszelkie diagramy umieszczane w dokumentacji należy zamieszczać tak aby były czytelne. Wymaganiem jest aby każdy dołączony diagram załączony do dokumentu posiadał swoją wersję w formacie XMI (możliwość importu do narzędzi modelowania) i HTML (możliwość przeglądu przez odbiorców nieposiadających oprogramowania specjalistycznego).

2. Dokumentacja Wymagań Biznesowych

Zakres dokumentacji określa standard S_04 „Standard Opisu Wymagań Biznesowych w Grupie Energia”.

3. Projekt Funkcjonalny (PF)

Celem Projektu Funkcjonalnego jest przedstawienie sposobu realizacji i działania systemu, abstrahując przy tym od szczegółów implementacyjnych, które zawarte zostaną w Projekcie Technicznym. Dokument zawiera dekompozycję funkcji systemu i ukazuje sposób ich działania.

Należy wyróżnić dwa typy dokumentu Projekt Funkcjonalny:

1. Projekt Funkcjonalny Zmiany – opisuje jedynie zakres wprowadzanych zmian do istniejącego systemu, a tym samym sposób realizacji określonego podzbioru wymagań biznesowych.
Wymaganiem jest aby dokument zawierał:
 - Metryka - autorzy, sprawdzający, daty przeglądów dokumentu, osoby je wykonujące, wprowadzone zmiany i ich źródło (historia zmian).
 - Osoba odpowiedzialna za utrzymanie aktualności dokumentu - wskazana osoba z imienia i nazwiska (w tym mail) odpowiedzialna za utrzymanie aktualności dokumentu.
 - Słownik pojęć i skrótów - użyte w dokumentacji definicje, akronimy i skróty wraz z jasnym i jednoznacznym objaśnieniem ich interpretacji
 - Ogólny opis zakresu wprowadzanych zmian wraz ze specyfikacją elementów systemu, które uległy modyfikacji wskazując referencje do funkcjonalności opisanych w Projekcie Funkcjonalnym Systemu.
 - Listę wymagań biznesowych objętych Projektem Funkcjonalnym.

- Opis realizacji wymagania – szczegółowy opis sposobu realizacji wymagania w systemie. Rekomendowanym jest aby niniejszy opis opierał się o makiety ekranów, przypadki użycia i diagramy UML (najczęściej przydatne diagramy: klas, sekwencji, czynności (aktywności), stanów). Tym samym zaleca się stosowanie powszechnie znanego zunifikowanego języka modelowania.
2. Projekt Funkcjonalny Systemu – opisuje pełen zakres systemu co oznacza, że należy go uznać za model analityczny rozwijanego/budowanego systemu. Stanowi bazę do opracowania Projektów Funkcjonalnych Zmian dla napływających żądań zmian do systemu.

Wymaganiem jest aby dokument zawierał:

- Metryka - autorzy, sprawdzający, daty przeglądów dokumentu, osoby je wykonujące, wprowadzone zmiany i ich źródło (historia zmian).
- Osoba odpowiedzialna za utrzymanie aktualności dokumentu- wskazana osoba z imienia i nazwiska (w tym mail) odpowiedzialna za utrzymanie aktualności dokumentu.
- Słownik pojęć i skrótów - użyte w dokumentacji definicje, akronimy i skróty wraz z jasnym i jednoznacznym objaśnieniem ich interpretacji
- Zakres i cechy produktu – ogólny opis specyfikowanego systemu i jego przeznaczenie.
- Materiały uzupełniające - wskazanie dokumentów i ich lokalizacji typu procedury, ustawy, rozporządzenia, zarządzenia etc. na których bazują wymagania oraz ich analiza.
- Opis ogólny – opis granic systemu, jego użytkowników i zależności zewnętrznych.
 - Perspektywa produktu - Kontekst systemu, ustalenie granic systemu, co jest systemem, a co jego środowiskiem (diagram kontekstowy).
 - Klasy i charakterystyka użytkowników.
 - Ograniczenia implementacji.
 - Założenia i zależności - Przyjęte założenia, zależności systemu/projektu od czynników zewnętrznych.
- Funkcje systemu – opis funkcji realizowanych przez system wraz z jednoznacznym objaśnieniem sposobu ich działania. Struktura opisu powinna zawierać:
 - Obszar A – logiczny podział na obszary/moduły w zależności od wielkości systemu (np. obszar Zarządzanie Umową):
 - Opis – ogólny opis obszaru.
 - Funkcjonalność X:
 - Historia zmian.
 - Identyfikator - unikatowy identyfikator funkcjonalności.
 - Artefakty powiązane – wykaz funkcjonalności powiązanych, np. założenie sprawy i utworzenie umowy.
 - Sekwencje bodziec/reakcja - sekwencje działań użytkowników i reakcji systemu, które definiują działanie danej funkcji.
 - Realizowane wymagania – identyfikatory realizowanych wymagań biznesowych.
 - Powiązane reguły biznesowe.
 - Wymaganie niefunkcjonalne dla funkcjonalności.
 - Uwagi i otwarte pytania
- Wymagania dotyczące danych:
 - Logiczny model danych – encje biznesowe, ich relacje i atrybuty; mapowania na wzorce dokumentów biznesowych, jeśli istnieją; identyfikator pola, opis, skład lub typ danych, typ pola, wymagalność, wartości (np. lista słownikowa - wskazanie na słownik), uwagi, reguły przetwarzania.
 - Raporty:
 - Raport X – unikatowy identyfikator raportu, tytuł raportu, cel raportu,, użytkownicy raportu, częstotliwość i dostępność raportu, źródła danych (mapowanie danych raportu na logiczny model danych), Typ pliku wynikowego(.docx, .xlsx, .pdf etc.), układ raportu (tj. definicja nagłówek, stopek, treści raportu (w tym kryteria wyboru i kryteria sortowania)), Ograniczenia dostępu związane z bezpieczeństwem (np. pracownik działu może uzyskać wyłącznie raport dla spraw ze swojego działu). Jeśli jest to

możliwe należy dołączyć wzór raportu ze wskazaniem danych z logicznego modelu .

- Integralność, przechowywanie, anonimizacja i usuwanie danych – określenie zasad i czasów po jakim mogą być usuwane dane z systemu.
- Wymagania dotyczące interfejsów zewnętrznych – wymagane dane i usługi przez i wobec otoczenia lub innych systemów.
- Wymagania dotyczące bezpieczeństwa.

4. Projekt Techniczny (PT)

Celem Projektu Technicznego jest przedstawienie technicznych aspektów projektowanego systemu przed jego wytworzeniem lub zmianą. Dokument stanowi na etapie powykonawczym składową dokumentacji technicznej.

Wymaganiem jest aby dokument zawierał:

- Metryka – autorzy, sprawdzający, daty przeglądów dokumentu, osoby je wykonujące, wprowadzone zmiany i ich źródło (historia zmian).
- Słownik pojęć i skrótów - użyte w dokumentacji definicje, akronimy i skróty wraz z jasnym i jednoznacznym objaśnieniem ich interpretacji Założenia niefunkcjonalne – zidentyfikowane założenia niefunkcjonalne mające znaczenie dla projektowanego systemu i dla których możliwe jest określenie wartości lub warunków brzegowych.

Przykładem takich założeń może być reżim operacyjności aplikacji/systemu (24/7, 16/7); liczba użytkowników; szacunkowa ilość jednoczesnych użytkowników w szczycie na godzinę; rodzaj dostępu użytkownika (Internet, użytkownik zewnętrzny); szacunkowy czas trwania pojedynczej sesji użytkownika; jakie i jak długo powinny być przechowywane dane w systemie; ilość danych przesyłanych od i do użytkownika podczas jednej typowej sesji (1-10MB rodzaj danych np. plik binarny obraz faktury); roczny przyrost danych; itp. itd.

- Projekt Techniczny Infrastruktury składający się z:
 - Architektura rozwiązania - ogólny opis i diagram rozwiązania w zakresie infrastruktury z uwzględnieniem infrastruktury już istniejącej lub wskazaniem konieczności inwestycji w nową infrastrukturę.
 - Specyfikacji sprzętu - zestawienie parametrów technicznych maszyn/serwerów rozwiązania (Nazwa maszyny/serwera, typ, typ procesora, ilość core, ilość RAM, przeznaczenie), z wyszczególnieniem wykorzystywanej infrastruktury istniejącej i koniecznej do zakupienia.
 - Opisu Warstwy bazodanowej - opis rodzaju i konfiguracji baz danych, ich wersji, klastrów bazodanowych, użytych systemów plików, parametry konfiguracyjne bazy danych, itp.
 - Opisu Warstwy aplikacyjnej - opis wymaganego oprogramowania wspomagającego (OS, middleware itp.), diagram rozmieszczenia na serwerach, mapowanie oprogramowania (np. bazy danych, serwerów aplikacyjnych, aplikacji) na infrastrukturę, itp.
 - Opisu Warstwy sieciowej (CPD/STGE/LAN) – planowana adresacja, planowane zmiany w topologii sieci, połączenia pomiędzy poszczególnymi elementami sieci, rodzaje połączeń, obciążenie łącz, wpływ na infrastrukturę sieciową, w tym na infrastrukturę obszaru bezpiecznego dostępu do Internetu, itp.
 - Opisu Systemu backupu - opis systemu backupu, opis polityki backupowej, wpływ na system backupu.
 - SAN - informacje o wymaganych przestrzeni pamięci masowej, wpływ na infrastrukturę SAN (Storage Area Network).

Np.

Oznaczenie	Model	Kontroler	Cache kontrolera	Interfejsy	Obsługa RAID	Dysk	Przestrzeń użytkowa [GB]
S-Storage-1	HP 3PAR 7200	2	12	FC 8GB	10, 50	450GB SAS	8500

- Specyfikacji zapotrzebowania na licencje i oprogramowanie wraz z uzasadnieniem i podziałem na środowiska, ze wskazaniem elementów krytycznych do prawidłowego funkcjonowania rozwiązania –od fizycznego sprzętu po middleware.

- Projekt Techniczny Systemu składający się z:
 - Opisu Komponentu Systemu - opis systemu od środka, jak dekomponuje się na komponenty, które komponenty odpowiadają za konkretne zadania, w jaki sposób realizowany jest mechanizm współdziałania komponentów, czy komponenty komunikują się z systemami zewnętrznymi.
 - Logicznego modelu danych - model danych implementowanego rozwiązania (wizualizacja poziomu logicznego).
 - Opisu Architektury bezpieczeństwa - opis zastosowanych rozwiązań zapewniających bezpieczeństwo systemu. Przykładowo zastosowanie szyfrowania transmisji danych, konfiguracja kluczy, Disaster Recovery Plan, zasady audytu (śledzenie czynności wykonywanych przez użytkowników), sposób uwierzytelniania, schemat ruchowy dla aplikacji (wszystkie planowane przepływy sieciowe w aplikacji z określonymi parametrami TCP/UDP/IP – zarówno klienckie, jak i dla ruchu wewnętrznego aplikacji, z określeniem aplikacji wykorzystujących dany przepływ).]
 - Projektu Techniczny Integracji - opis współpracy z innymi systemami, sposób realizacji interakcji.
 - Opis podejścia do integracji – diagram obrazujący komunikację z otoczeniem
 - Lista interfejsów – unikatowy identyfikator, Nazwa interfejsu, dostawca i konsument usługi.
 - Specyfikacja interfejsów – każda z usługi opisana wg standardu dokumentowania interfejsów (kontrakty usług).
 - Specyfikacji zapotrzebowania na licencje i oprogramowanie, ze wskazaniem elementów krytycznych do prawidłowego funkcjonowania rozwiązania - lista wymaganych licencji (rodzaj licencji, liczba) na oprogramowanie, czy biblioteki itp. niezbędne do realizacji zakresu przez programistów, np. licencje na biblioteki OCR, licencje na framework ORM, itp. Zapotrzebowanie na moc obliczeniową i licencje generowane na poziomie od fizycznego sprzętu po middleware opisujemy w części Projektu Technicznego dotyczącego Infrastruktury.
- Specyficzne warunki i ograniczenia - w tym:
 - Problemy i ograniczenia technologiczne w oprogramowaniu COTS - lista znanych ograniczeń i błędów związanych z oprogramowaniem, które wpływają na wybór architektury (problem, opis szczegółowy, alternatywa/sposób obejścia).

5. Dokumentacja Techniczna (DT)

Dokumentacja Techniczna opracowywana jest w celu umożliwienia, sprawnego przeprowadzenia operacji w postaci: przygotowania środowisk deweloperskich, modyfikacji ich konfiguracji, kompilacji kodu, instalacji/konfiguracji i uruchomienia systemu IT.

Wymaganym jest aby dokumentacja Techniczna zawierała:

- Metryka - autorzy, sprawdzający, daty przeglądów dokumentu, osoby je wykonujące, wprowadzone zmiany i ich źródło (historia zmian).
- Słownik pojęć i skrótów - użyte w dokumentacji definicje, akronimy i skróty wraz z jasnym i jednoznacznym objaśnieniem ich interpretacji.
- Projekt Techniczny - zawartość projektu opisano w rozdziale 4 niniejszego dokumentu..
- Dokumentacja kodu źródłowego i bazodanowego¹ - np. wygenerowana dokumentacja (np. javadoc) lub wskazanie jej lokalizacji, raport html schematów bazodanowych wykorzystywanych przez aplikacje, itp.
- Instrukcja konfiguracji środowiska deweloperskiego - szczegółowa procedura konfiguracji środowiska deweloperskiego.
- Procedura kompilacji/budowy aplikacji i deploymentu¹ - szczegółowa procedura kompilacji/budowy aplikacji i deploymentu, w tym opisy najczęstszych problemów i ich rozwiązań/obejść.
- Rozmieszczenie poszczególnych pakietów oprogramowania – np. w formie diagramu wdrożenia (UML).
- Opis konfiguracji systemu - opis dodatkowych czynności konfiguracyjnych np. do wykonania przed/po deploymentie aplikacji, do poprawnego działania systemu.

¹ Punkt opcjonalny dla oprogramowania typu COTS wytwarzanego wewnątrz Grupy.

- Kontrakty Usług - dokumentacja udostępnianych usług przygotowana zgodnie ze standardem opisanym w dokumencie „Standard Dokumentowania Interfejsów w Grupie Energa”, wskazana forma oddzielnych dokumentów dla każdej z usług.

6. Dokumentacja Administratora (DA)

Wymaganym jest aby dokument Dokumentacja Administratora zawierał:

- Metryka - autorzy, sprawdzający, daty przeglądów dokumentu, osoby je wykonujące, wprowadzone zmiany i ich źródło (historia zmian).
- Słownik pojęć i skrótów - użyte w dokumentacji definicje, akronimy i skróty wraz z jasnym objaśnieniem ich interpretacji.
- Wykaz lokalizacji najważniejszych plików eksploatacyjnych - lokalizacja logów poszczególnych komponentów, plików konfiguracyjnych itp.
- Procedury monitorowania i utrzymania - opis sposobu monitorowania działania systemu i jego konserwacji, opis standardowych czynności administracyjnych.
- Indeks błędów - zestawienie możliwych błędów, na które może natknąć się administrator, w wraz z opisem i koniecznym działaniem w celu ich usunięcia.
- Opis konfiguracji poszczególnych elementów systemu

7. Dokumentacja Użytkownika (DU)

Celem dokumentu jest opisanie sposobu korzystania z aplikacji przez użytkowników końcowych. W dokumencie powinny być opisane wszystkie funkcje użytkowe wchodzące w skład aplikacji.

Wymaganym jest aby dokument zawierał:

- Metryka - autorzy, sprawdzający, daty przeglądów dokumentu, osoby je wykonujące, wprowadzone zmiany i ich źródło (historia zmian).
- Zakres - ogólny opis zakresu dokumentu, jego rozdziałów.
- Konwencje stosowane w dokumencie - opis zastosowanych konwencji w dokumencie, ułatwiający odczytanie zawartych w nim informacji (np. sposób oznaczenia w tekście pola edycyjnego, elementu sterującego, klawiszy, itp.).
- Słownik pojęć i skrótów - użyte w dokumentacji definicje, akronimy i skróty wraz z jasnym i jednoznacznym objaśnieniem ich interpretacji.
- Opis elementów sterujących interfejsu użytkownika - wygląd i opis np. pole edycyjne, lista wyboru, przycisk, pasek menu, pole wyboru (checkbox/radio button), kalendarz, grupa pól itp.
- Charakterystyka aplikacji X - krótki opis wprowadzający o aplikacji, której elementy są opisywane w dalszej części dokumentu.
- Uprawnienia – opis modelu uprawnień, macierz uprawnień (rola vs funkcjonalność, do której ma dostęp).
- Uruchomienie systemu X - opis uruchomienia aplikacji.
- Charakterystyka modułu A systemu X – krótki opis wprowadzający o module aplikacji, którego elementy są opisywane w dalszych rozdziałach.
 - Opis funkcji udostępnianych użytkownikowi - opis funkcji modułu, sposobu korzystania i efektu użycia.
 - Indeks błędów - zestawienie możliwych błędów, na które może natknąć się użytkownik, ich komunikatów wraz z opisem i sugerowanym działaniem użytkownika w celu ich usunięcia, czy ominięcia.

Standard Dokumentacji Interfejsów w Grupie Kapitałowej Energia

Wstęp

Niniejszy dokument powstał w związku z wprowadzaniem Katalogu Standardów IT wynikającego z Kodeksu Architektury GK Energia będącego częścią Polityki IT Grupy Energia.

Celem dokumentowania interfejsów jest utwalenie i dostarczanie wiedzy na temat oferowanych usług, dotyczącej między innymi opisu jak używać dostępne usługi, jak uzyskać dostęp od usługi, kiedy i jakie błędy mogą wystąpić, jakie wykorzystano standardy przy budowie integracji, jakie występują ograniczenia, jaki jest praktyczny przykład użycia (wywołania) operacji interfejsu.

Uregulowania zawarte w Standardzie Dokumentacji Interfejsów w Grupie Kapitałowej Energia obowiązują każdą jednostkę organizacyjną realizującą czynności związane z rozwojem i/lub utrzymaniem oprogramowania.

1. Definicja Interfejsu

Interfejs, inaczej Usługa Sieciowa, to funkcjonalność systemu informatycznego wykonującą z góry określone funkcje w odpowiedzi na otrzymane za pomocą sieci teleinformatycznej uporządkowane w określonej strukturze dane. Interfejs posiada określony zakres funkcjonalny i składa się z operacji.

2. Zakres dokumentacji interfejsu

Dopuszczalne jest dokumentowanie API ²systemu przy użyciu narzędzi automatycznego dokumentowania jako uzupełnienie tworzonej dokumentacji.

W przypadku interfejsów implementowanych na szynach danych (ESB) dokumentacja powinna opisywać tylko obszar zaimplementowany w ich strukturach i posiadać dołączoną dokumentację interfejsów (lub referencje do systemów udostępnianych, tak by tworzyły one spójną całość.

Dla podmiotów zewnętrznych, którym udostępnia się usługi np. bramkę B2B należy utworzyć dodatkowe instrukcje użycia, nieujawniające wewnętrznej konstrukcji rozwiązania zgodnie z modelem black box.

Wymaganym jest, aby dokumentacja interfejsu zawierała takie informacje jak:

- Nazwa interfejsu - unikatowa nazwa interfejsu.
- Metryka - autorzy, sprawdzający, daty przeglądów dokumentu, osoby je wykonujące, wprowadzone zmiany i ich źródło (historia zmian).
- Osoba odpowiedzialna za utrzymanie aktualności dokumentacji interfejsu - wskazana osoba z imienia i nazwiska (w tym adres mail oraz numer telefonu służbowego) odpowiedzialna za utrzymanie aktualności dokumentu.
- Przeznaczenie interfejsu – opis w jakim celu jest on wykorzystywany.
- Słownik pojęć i skrótów - użyte w dokumentacji definicje, akronimy i skróty wraz z jasnym i jednoznacznym objaśnieniem ich interpretacji.
- Specyfikacja usługi:
 - Wykorzystane standardy do budowy usługi.
 - Adres usługi.
 - Warunki początkowe do spełnienia - informacja na temat warunków początkowych interfejsu np. konieczność zestawienia tunelu, uwierzytelnianie, autoryzacja, etc. wraz z opisem sposobu ich spełnienia.
 - Specyfikacja usługi - specyfikacja WSDL lub inna użyta do utworzenia usługi.
 - Lista operacji usługi - lista operacji wraz z krótkim opisem sposobu ich funkcjonowania.

² API – Application programming interface – interfejs programowania aplikacji

- Operacja <<nazwa-operacji>> - szczegóły każdej z operacji usługi zawierające co najmniej opis składni oraz użytej semantyki:
 - Opis ogólny - ogólny opis przeznaczenia i roli operacji.
 - Opis szczegółowy działania operacji, w tym:
 - Dane wejściowe - opis wejścia operacji, w tym wskazanie typów danych, stałych, wymagalności, kluczy głównych, kodów - precyzując przy tym ich znaczenie.
 - Dane wyjściowe - opis wyjścia operacji, w tym wskazanie typów danych, stałych, kody precyzując przy tym ich znaczenie, oddzielny opis kodów błędów i ich możliwych przyczyn wystąpienia.
 - Efekt użycia operacji - opis zdarzeń niewidocznych z perspektywy klienta usługi np. zmiany stanów obiektów po stronie dostawcy usługi, które są ważne z punktu widzenia klienta oraz samej operacji.
 - Biznesowy opis zachowania - biznesowy opis sekwencji wykonywanych przez konkretną operację.
 - Przykład użycia - pełen przykład użycia (wywołania) usługi, wraz z przykładowymi komunikatami wejściowymi i wyjściowymi, dodatkowym komentarzem etc.
 - Obsługa błędów - opis kodów błędów i ich możliwych przyczyn wystąpienia – jeśli liczba błędów jest mała wystarczy opis zawarty przy Danych Wyjściowych operacji.
- Ograniczenia dla usługi - wszelkiego rodzaju ograniczenia, w tym wydajnościowe, czy SLA.
- Kwestie projektowe - rozdział utrwalający wiedzę na temat motywacji wyboru technologii, architektury oraz wzorca zaprojektowanego interfejsu, poczynionych kompromisów, rozpatrywanych i odrzuconych alternatyw wraz z przyczyną odrzucenia oraz wszelkie informacje, które w przyszłości należy uwzględnić podczas projektowania nowej wersji usług.
- Inne informacje ważne z punktu widzenia danego interfejsu - rozszerzenie dokumentacji o specyficzne i wartościowe informacje dla odbiorców dokumentacji.

Standard Wytwarzania Oprogramowania w Grupie Kapitałowej Energia

Wstęp

Niniejszy dokument powstał w związku z wprowadzaniem Katalogu Standardów IT wynikającego z Kodeksu Architektury GK Energia będącego częścią Polityki IT Grupy Energia.

Celem standaryzacji wytwarzania oprogramowania jest wskazanie elementów, które powinny być składową cyklu produkcji oprogramowania na potrzeby Spółek Grupy Energia, w celu bezpiecznego i efektywnego utrzymania i rozwoju danego systemu IT.

Uregulowania zawarte w Standardzie Wytwarzania Oprogramowania w Grupie Kapitałowej Energia obowiązują każdą jednostkę organizacyjną odpowiedzialną za realizację wytwarzania oprogramowania.

1. Definicja wytwarzania oprogramowania

Wytwarzanie oprogramowania w niniejszym standardzie rozumiane jest jako proces implementacji (tworzenie kodu źródłowego oprogramowania) określonych wymagań biznesowych, funkcjonalnych i нефункциональных i kontroli jakości wytwarzanego kodu.

2. Wytyczne standardu wytwarzania oprogramowania

Wymaganiem jest aby wytwarzanie oprogramowania na potrzeby Grupy Energia realizowane było w sposób uwzględniający następujące oczekiwania:

1. Repozytorium kodu źródłowego (SVN, GIT lub inne dostarczane przed producenta danego oprogramowania COTS) umiejscowione było na infrastrukturze Grupy Energia.
2. Repozytorium kodu źródłowego powinno zawierać wszelkiego rodzaju skrypty wytworzone podczas implementacji rozwiązania (np. skrypty budujące fizyczny model danych w bazie).
3. Zespoły Wytwarzające oprogramowanie powinny utrzymywać i stosować instrukcje zawierające wytyczne dotyczące struktury repozytorium kodu źródłowego w szczególności opisujące zasady tworzenia, nazewnictwo i przeznaczenie katalogów i gałęzi, tagowanie i zasady commitowania zmian (bez ograniczenia do).
4. Dokumentacja systemu IT przechowywana jest w ramach repozytorium dostępnego dla wszystkich uczestników procesu wytwarzania oprogramowania.
5. Wszystkie tworzone obiekty bazodanowe (w szczególności tabele i ich kolumny) powinny być opatrzone komentarzami (np. w Oracle RDBMS: "comment on column..."), jeśli tylko jest taka możliwość (np. brak ograniczeń licencyjnych, czy technicznych).
6. Klasy i ich metody powinny być opatrzone stosownymi komentarzami wyjaśniającymi, ułatwiającymi późniejsze wykorzystanie, czy modyfikację.
7. Dla newralgicznych obszarów kodu należy implementować testy jednostkowe i automatyczne, minimalizujących ryzyko niepoprawnego działania oprogramowania i ułatwiających testy m.in. regresyjne.
8. Zalecane jest korzystanie z metodyk typu Continuous Inspection, Continuous Integration, Continuous Deployment itp.
9. Zespoły Wytwarzające oprogramowanie stosują następujące zasady komentowania kodu źródłowego składającego się na Oprogramowanie:
 - 9.1. Wszystkie elementy kodu źródłowego (stałe, zmienne, instrukcje sterujące, pola klas, klasy, metody, funkcje, procedury, pakiety, biblioteki itp.) powinny zostać opatrzone komentarzem zawierającym biznesowy opis funkcjonowania.
 - 9.2. Każdy element taki jak pakiet, klasa, metoda, procedura, funkcja itp. zawiera opis nagłówkowy, zawierający informacje:
 - dane autora,
 - numer wersji,
 - data utworzenia i data ostatniej modyfikacji,
 - lista dokonanych zmian (kto, kiedy, co),
 - lista i opis argumentów (jeśli takie posiada),

- opis zwracanej wartości (jeśli zwraca wartość) lub wyniku działania,
 - wyczerpujący opis działania / słowny opis użytego algorytmu,
 - ewentualnie odwołanie do dokumentacji Oprogramowania.
10. Zespoły Wytwarzające oprogramowanie powinny utrzymywać i stosować się do wytycznych formatowania kodu i konwencji nazwicznych stosowanych w kodzie. Opracowane wytyczne będą zgodne z założeniami:
- 10.1. Zachowania konwencji nazewnictwa dla:
- wszystkich elementów kodu (pakiety, biblioteki, klasy, metody, pola klas, stałe, zmienne, funkcje, procedury itp.),
 - wszystkich składników systemu baz danych (baza danych, tabele, kolumny),
 - innych składników Oprogramowania.
- 10.2. Kod powinien być sformatowany w jednolity sposób.
11. Wyjątki powinny być obsługiwane wewnątrz funkcji, a jeśli jest to niewskazane z przyczyn konstrukcyjnych programu, to lista możliwych do zwrócenia wyjątków musi być zadeklarowana i odpowiednio opisana.
12. Dla Oprogramowania pisanego w języku java klasy i metody powinny być opisane sposób pozwalający na wygenerowanie dokumentacji za pomocą narzędzia javadoc.
<http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>
13. Wszystkie operacje zmieniające zawartość danych w bazach danych będą wykonywane transakcyjnie; w kodzie źródłowym zostanie zaimplementowana obsługa ewentualnych błędów wykonania operacji na bazie danych.
14. Wszelkie funkcje na bazie danych będą zaimplementowane w taki sposób, iż dołożenie jakiegokolwiek kolumny do dowolnej tabeli w bazie danych, zmiana kolejności kolumn w jakiegokolwiek tabeli w bazie danych, zmiana definicji wielkości kolumny tekstowej na większą ilość znaków lub zmiana definicji kolumny na przechowywanie większych wartości numerycznych nie będzie wpływała na działanie Oprogramowanie.
15. W kodzie źródłowym musi zostać obsłużona funkcjonalność zapisu błędów i wyjątków w kontekście zdarzeń biznesowych do dziennika zdarzeń (loga).