

Uwagi dotyczące projektów z Baz Danych:

[Proszę przeczytać wszystko i wybrać to co stosuje się do projektu tworzonego przed Panie/Panów]

Konceptualny: do 5 stron!

Logiczny: do 10 stron, zawiera konceptualny.

Całość: do 15 stron!!!

Ale są wyjątki.

Strona tytułowa (projekty „papierowe”) lub strona główna projektu w Wiki – proszę zawsze podawać:

- temat projektu,
- rok, rok akademicki, miesiąc,
- faza: konceptualny, logiczny, całość = projekt końcowy,
- skład grupy,
- **kontakt: e-mail.**

Uwaga: to można zrobić na 1/3 lub 1/4 strony u góry i od razu pisać - wykorzystać resztę strony/miejsce na treść projektu.

1. Tytuł: proszę nie zapominać o podaniu tytułu/tematu projektu!!!.

2. Analiza stanu zastanego/wyjściowego zawiera nie tylko ew. poprzedni system papierowy/informatyczny, ale:

- obieg i formaty dokumentacji papierowej,
- uregulowania prawne, przepisy, zwyczaje lokalne, etc.,
- wymagania specyficzne techniczne zastane (sprzęt, syst. operacyjny, software, sieć,...),
- istotne ograniczenia (np. czas obsługi systemu < 10 sek., szybki interfejs tekstowy, obsługa bez myszy, kolory/brak),
- wymagania niezawodnościowe,
- wymagania bezpieczeństwa,
- obsługa innych urządzeń: fax, skaner, OCR, kamera, drukarki, czytnik kart,
- formaty wydruku + typy plików,
- koszty, licencje, koszty oprogramowania, koszty utrzymywania, koszty administracji,
- analizę organizacji systemu, prac, dostępu,
- możliwość modyfikacji, rozszerzeń,
- kwestia wielojęzyczności,
- inne.

Te aspekty mogą i powinny pojawiać się także w analizie wymagań.

W tej części można i należy oprzeć się na literaturze i innych źródłach – np. internecie. Zredagować listę i cytować numerkami.

3. Analiza wymagań użytkownika: często dzieli się je na jawne i niejawne; obejmuje:

- opis podstawowych wymagań co do **funkcjonalności** systemu,
- wymagania poza funkcjonalne, różne ich kategorie:
- wymagania techniczne: sprzęt, software, technikalnia,

- wymagania formalne typu informatycznego: kodowanie znaków, język(i), możliwość reprezentacji danych różnych typów (np. obrazki), współpraca z innymi systemami, możliwości importu i eksportu danych, dostęp przez internet, obsługiwane formaty plików, etc.
- wymagania ekonomiczno-organizacyjne (koszty, legalność),
- kwestie przenaszalności, utrzymywalności, rozszerzalności, (systemy z administratorem lub bez), wymagania jakościowe i niezawodnościowe, interoperacyjność, etc.).
- zgodność z aktami prawa i lokalnymi dokumentami, zwyczajami, etc.,
- problematyka bezpieczeństwa (każdy wymiar),
- istotne ograniczenia – jeżeli występują.

4. Scenariusze użycia: to przykładowe "historyjki" co i jak się dzieje, są podstawą do analizy ukierunkowanej na określenie realizowanych funkcji, sterowania, obiektów, atrybutów, etc. Mogą pochodzić z instrukcji obsługi, obserwacji pracy, etc.

Scenariusze użycia przypominają swoiste procedury dostępne w systemie i realizujące wymagane i sprawdzone sposoby dostępu do danych.

Np. b. ważne dla metra – jak będziemy się posługiwać systemem – procedury. Na podstawie scenariuszy użycia określa się funkcjonalność systemu.

5. Identyfikacja funkcji: na podstawie scenariuszy – wstępna, nieformalna, uzgodniona z użytkownikiem. Zakresy/obszary funkcjonalności.

W bazach danych: selektywny dostęp, insert, update, delete. Kto, jaki zakres.

6. Diagramy FHD: mają strukturę hierarchiczną! To oznacza, że funkcje niższego poziomu, np. 1.1, 1.2, 1.3, są elementami składowymi czynności wyższego poziomu, określonej bardziej ogólnie, nr 1. Nie określają kolejności, związków przyczynowo-skutkowych, etc. Zawsze powinny pokrywać całość funkcjonalności (wymóg kompletności), a poszczególne funkcje powinny być rozróżnialne (rozłączne).

Przy definiowaniu funkcji proszę pamiętać, że:

- wszelkie dane muszą się jakoś znaleźć w systemie - ktoś musi je wprowadzać (lub są wprowadzane automatycznie)!
- dane podlegają zwykle czterem operacjom: wyszukiwanie, wprowadzanie, aktualizacja, kasowanie; oczywiście są inne operacje szczegółowe, proszę też pamiętać o raportach/agregacji.
- wyszukiwanie powinno być możliwie elastyczne - proszę pomyśleć o różnorodnych kryteriach!!! I pamiętać: podstawową funkcją każdej bazy danych jest zapewnienie SELEKTYWNEGO dostępu do danych!!!

7. Diagramy DFD: proszę o diagram kontekstowy (poziom 0), diagram poziomu 1 i diagram poziomu 2, ew. 3. Wszystkie funkcje na danym diagramie są tego samego poziomu, a więc nie można mieszać 1.2 z 1.2.3 - 1.2.3 jest elementem składowym 1.2.

Proszę numerować funkcje - tak jak zostały ponumerowane w FHD i także je nazywać.

8. Określenie encji i atrybutów: proszę krótko opisać, co niejasne. Wybór

encji i atrybutów – to też ważny krok; proszę pamiętać, że raz określone atrybuty pozostają na lata w strukturze bazy. Niektóre grupy zapomniały.

9. Diagramy ERD: też wymagają omówienia. Dobrze, aby było widoczne co z czym jest łączone. Ładny layout, bez przecinania/pokrywania linii.

10 Diagramy STD: to mapa współpracy z systemem. Jedna strona www może realizować wiele funkcji. Pamiętać o przejściach/powrotach i zachowaniu danych. Na diagramie można posłużyć się numerami z FHD.

Proszę pamiętać, że diagramy też wymagają omówienia: znaczenie symboli, nazw, napisów, etc. I sama konstrukcja diagramu. Dodatkowo: wyjaśnienie relacji, przejść, etc.

Logowanie: typowym problemem jest niezapamiętywanie loginu i hasła. Proponuję rozważyć login jako e-mail (jest chyba jednoznaczny), daje możliwość przypominania hasła na e-mail.

Na rysunkach należy rozplanować położenie elementów! Napisy nie mogą nakładać się na elementy graficzne (wyliminować kreślenie).

Mile widziane inne uzupełnienia projektu – choćby szkicowe. Np. kopie analizowanych dokumentów, odwołania do przepisów, analiza innych podobnych systemów, linki do ciekawych stron (teoria, projekty, narzędzia). Narzędzia wspomagające.

Polecam program do wizualizacji – dla tych co nie lubią rysować:

<http://www.graphviz.org/> (są tam też diagramy ERD (niestety stare) i datastruct – mogą udawać ERD. Ponadto dbDesigner. Oraz Dia i Xfig.

Poszukajcie też w Gnome i KDE – są różne nowości. Najciekawsze znaleziska (np. Umbrello) będą premiowane.

Projekt logiczny:

11. Kod SQL – tworzenie bazy:

To jest kluczowa część projektu!

Proszę pamiętać o:

- ustaleniu kodowania (latin2 lub utf-8); kodowanie serwera i klienta,
- ustaleniu formatu daty,
- odpowiednim doborze typów danych i rozmiarów.

Ciągle zdarzają się definicje typu PESEL lub telefon jako integer! Albo numery domów, lokali, etc. Z drugiej strony 40 znaków na kod pocztowy jest też przesadą.

- w wielu pracach zabrakło definicji kluczy obcych; obok właściwej definicji kluczy to najważniejsza część dotycząca więzów integralności, a więc zapewnienia poprawności projektu!!!
- do rzadkości należało wykorzystanie opcji DEFAULT a zwłaszcza CHECK

- proszę nie pisać NULL (to oczywiste, default); deklarować tylko NOT NULL (i to ostrożnie, tam gdzie trzeba).
- użycie definicji DOMAIN (dziedziny w PostgreSQL), sekwencje (SEQUENCE) i CONSTRAINT to już absolutna rzadkość – proszę z nich też korzystać!
- nie szafować niepotrzebnie rozmiarami varchar – często stosowanie rozmiaru 255 jest bezsensowne.

Plik drukować środowiskiem verbatim lub podobnym (fancyvrb), ale:

- plik przeniesiony 1:1 z "chodzącej wersji" zdjętej z systemu.
- komentarze jeżeli jest coś niejasnego.

Można użyć środowiska listings dla LaTeX'a lub <code sql> w Wiki; przykładowa kombinacja dla LaTeX'a komend poniżej.

```
\usepackage{listings}
\usepackage{color}
\
lstset{language=SQL,basicstyle=\normalsize\ttfamily,keywordstyle=\sffamily\
bfseries,stringstyle=\ttfamily,showstringspaces=false}

\begin{lstlisting}
SELECT ID_prac, Nazwisko, Data_ur
      FROM prac Z
      WHERE Data_ur <
            (SELECT Data_ur
              FROM prac W
              WHERE W.Dzial = Z.Dzial AND
                    W.Stanowisko='kierownik')
      ORDER BY ID_prac;

\end{lstlisting}
```

12. Słowniki danych.

Wiele osób zapomniało o tym. Proszę podawać słowniki tam gdzie to jest istotne, np. lista możliwych opcji, lista województw, lista kodów, dopuszczalne nazwy, specyficzne własne konstrukcje klucza, etc.

13, 14 – chwilowo pomijamy. Będzie na wykładzie.

15. Operacje.

Chodzi o ciekawe przykłady operacji. Może to być sam SQL lub kod SQL zanurzony w PHP.

Należy podać, jaką funkcję/funkcje (nr z FHD+nazwa) dany kod realizuje.

Chodzi o ciekawsze przykłady: zaawansowane wyszukiwanie, agregacja, podzapytania, złączenia, etc. Oczywiście transakcje, triggerzy, etc. Kod PL/SQL – także.

Ważne:

Cały projekt powinien być spójny!!!

A więc np. zadeklarowane w FHD/DFD funkcje powinny być widoczne w STD i opisane w p. 15 (choćby częściowo). Funkcje w p. 15 to właśnie wypełnienie

zobowiązań zadeklarowanych w FHD/DFD. W raporcie całościowym mile widziane zrzuty ekranu dokumentujące daną funkcjonalność.
Interfejs powinien być prosty – ale funkcjonalny i w miarę jednolity.

Ważne:

Wiele osób chyba nie czytało pierwszej części uwag – wiele rzeczy jest niepoprawionych, np. interpunkcja!

Raport końcowy obejmuje punkty do 20 (od 21 – tylko jeżeli ktoś chce i ma doświadczenia praktyczne z wdrażania, etc.).

Proszę podać wykaz literatury, użytych narzędzi (serwer BD, języki, programy graficzne do projektowania, etc.).