Zespołowy projekt CASE, etap projektowania

W fazie projektowania dużą role odgrywa środowisko implementacji. Projektanci muszą więc posiadać dobrą znajomość języków, bibliotek i narzędzi stosowanych w trakcie implementacji.

Generalnie, powinno być dążenie do tego, aby struktura projektu zachowała ogólną strukturę modelu stworzonego w fazie analizy. Dobrze by było, gdyby ewentualne niewielkie zmiany w dziedzinie problemu implikowały niewielkie zmiany w projekcie. Projektanci, wykonując zadania danego etapu, powinni wykorzystywać idei programowania strukturalnego i obiektowego.

Etap projektowania uszczegóławia wyniki analizy. Projekt musi być na tyle szczegółowy, aby mógł być podstawą implementacji. Stopień wymaganej szczegółowości zależy od poziomu zaawansowania programistów (koderów).

Głównym dokumentem etapu projektowania jest Dokumentu Detalicznego Projektu (DDP). Jest to szczegółowy opis rozwiązania problemu określonego w dokumencie wymagań na oprogramowanie.

Cel etapu

Opracowanie szczegółowego opisu (planu) implementacji systemu.

Zadania do wykonania

- Wybrać narzędzia do szybkiego rozwijania aplikacji (ang. RAD - Rapid Application Development). Wybór uzasadnić w krótkiej notatce tekstowej.
- Zaprojektować składowe systemu nie związane z dziedziną problemu, w tym interfejsy i składowe zarządzania danymi (wybrać technikę przechowywania trwałych danych aplikacji). W wyniku prac projektowych powinny powstać właściwe do wybranej metody diagramy, takie jak np. projektowy diagram klas, fizyczny model bazy danych, diagram komponentów itd.
- Dostosować projekt do ograniczeń i możliwości wybranego środowiska implementacji.
- Przeprowadzić optymalizację systemu.
- Określić fizyczną strukturę tworzonego systemu.

Wymagana dokumentacja etapu

1. Notatka tekstowa uzasadniająca wybór narzędzi RAD.
2. Dokument detaliczny projektu (DDP).

Źródła wiedzy

1. Projektowanie aplikacji OOP - w czym i jak? - http://flab.pl/forum/t/752/
2. Organizacja projektu, co najpierw, a co później - http://forum.php.pl/index.php?showtopic=38156
3. RAD – http://pl.wikipedia.org/wiki/RAD
4. Tworzenie aplikacji międzyplatformowych za pomocą narzędzi RAD firmy Borland - http://www.borland.pl/tech/interplatform.shtml
5. Dreamweaver MX, a rapid web application development tool supporting PHP – http://www.adobe.com/devnet/dreamweaver/.
6. Maciej Węgorkiewicz, Darmowy RAD dla Javy - JBuilder Foundation 3, PCkurier 4/2000 - http://www.pckurier.pl/archiwum/art0.asp?ID=3826

7. Bartosz Bębel, Materiały dydaktyczne do przedmiotu „Projekty informatyczne”, projektowanie w oparciu o środowisko Oracle Designer - http://www.cs.put.poznan.pl/bbebel/dydaktyka.html#MAT_PI
8. Maciej Węgorkiewicz, seria artykułów w PCKurier (w nr 7-13, 1999) na temat projektowania aplikacji w Visual Studio 6 - http://www.pckurier.pl/archiwum/
9. Michał Wolski, Projektowanie i kodowanie aplikacji w zintegrowanym środowisku programistycznym, Microsoft Visual Studio .NET oraz IBM Rational XDE – http://www.uml.com.pl/modules/articles/article.php?id=13
10. Marek Olejniczak, Wieloplatformowe środowiska i narzędzia programistyczne – http://olmar.poznan.pl/artykuly/uniplatform/uniplatform.html

Porady, wskazówki i podpowiedzi

- Zwrócić uwagę na to, że dostępne są różne metodologie rozwijania projektu: proceduralna, strukturalna, obiektowa, zdarzeniowa, agentowa, funkcyjna, liniowa, logiczna, deklaratywna, generyczna.
- DDP jest centralnym dokumentem, w którym zgromadzone są wszystkie informacje odnośnie budowy i działania oprogramowania. DDP powinien być zorganizowany w taki sam sposób, w jaki zorganizowane jest oprogramowanie. DDP powinien być kompletny, odzwierciedlający wszystkie wymagania zawarte w specyfikacji wymagań. Materiał, który nie mieści się w podanej zawartości dokumentu, powinien być załączony jako dodatek.
  DDP musi uwzględniać wszystkie wyspecyfikowane wymagania. Powinien być wystarczająco detaliczny aby umożliwić implementację i pielęgnację kodu. Styl DDP powinien być systematyczny i rygorystyczny. Język i diagramy użyte w DDP powinny być klarowne. Dokument powinien być łatwo modyfikowalny. Struktura DDP powinna odpowiadać strukturze projektu oprogramowania. Język powinien być wspólny dla całego dokumentu. Wszystkie użyte terminy powinny być zdefiniowane i użyte w zdefiniowanym znaczeniu. Zalecana zawartość DDP:

  Informacja organizacyjna

  Streszczenie

  Spis treści

  Formularz statusu dokumentu

  Zapis zmian w stosunku do ostatniej wersji

  OPIS OGÓLNY

  1. WPROWADZENIE

     1.1. Cel DDP

     1.2. Zakres DDP

     1.3. Definicje, akronimy, skróty

     1.4. Odsyłacze

     1.5. Krótkie omówienie treści dokumentu

  2. STANDARDY PROJEKTU, KONWENCJE, PROCEDURY

     2.1. Standardy projektowe

2.2. Standardy dokumentacyjne

2.3. Konwencje nazwowe

2.4. Standardy programistyczne

2.5. Narzędzia rozwijania oprogramowania

SPECYFIKACJA KOMPONENTÓW

n [IDENTYFIKATOR KOMPONENTU]

n.1. Typ

n.2. Cel

n.3. Funkcja

n.4. Komponenty podporządkowane

n.5. Zależności

n.6. Interfejsy

n.7. Zasoby

n.8. Odsyłacze

n.9. Przetwarzanie

n.10. Dane

Dodatek A. Listing kodu źródłowego.

Dodatek B. Macierz zależności pomiędzy zbiorem wymagań i zbiorem komponentów oprogramowania.