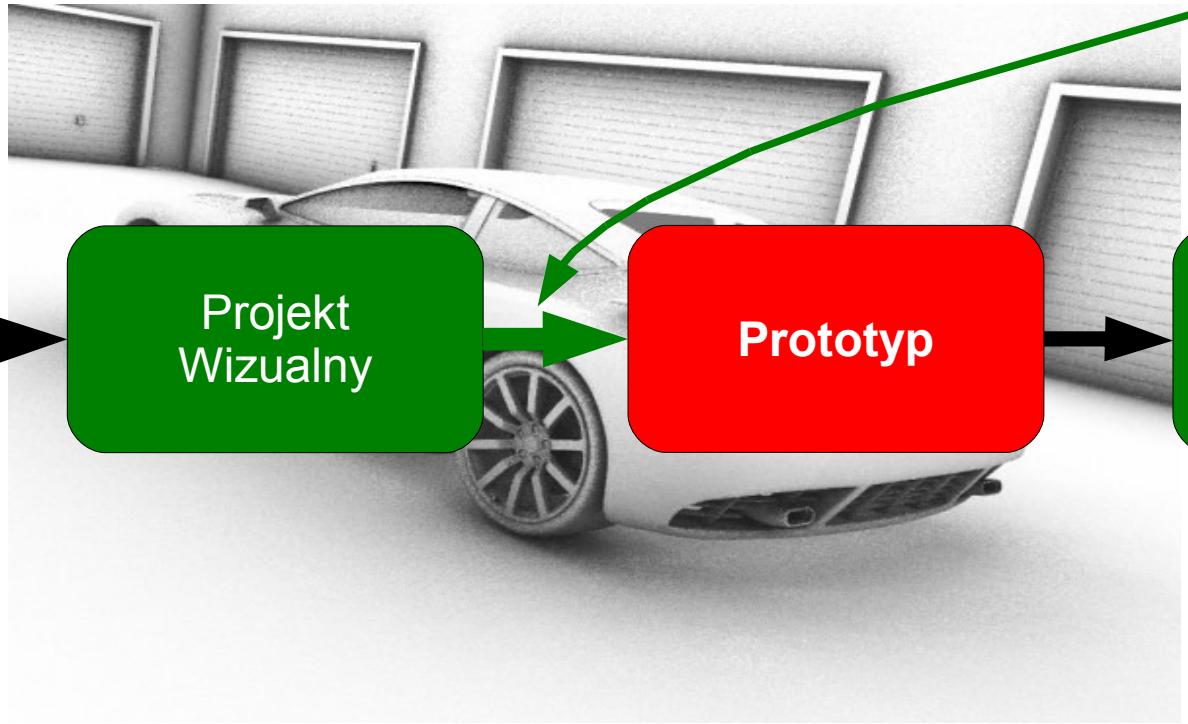
Inżynieria oprogramowania

Robert Szmurło

Prototypowanie + RAD

jako środek wytwarzania (implementacji)



Analiza → Projekt Wizualny → **Prototyp** → Zrealizowany System

Z http://www.blender.pl/cpg/albums/userpics/10505/normal_CarPrototypeConcept1b.jpg

Modele procesów tworzenia systemów inf.

Na podstawie: Ian Sommerville, „Inżynieria oprogramowania”, Rozdział 3

Model kaskadowy

- W tym modelu podstawowe czynności specyfikowania, tworzenia, zatwierdzania i ewolucji są odrębnymi fazami procesu takimi jak specyfikowanie wymagań, projektowanie oprogramowania, implementacja, testowanie itd.

Tworzenie ewolucyjne (związane z RAD i JAD)

- W tym procesie czynności specyfikowania, projektowania i zatwierdzania przeplatają się. Pierwsza wersja systemu powstaje bardzo szybko na podstawie abstrakcyjnych specyfikacji. Później jest udoskonalana zgodność z informacjami otrzymanymi od klienta.

Tworzenie formalne systemu (modelowanie)

- To podejście jest oparte na budowaniu formalnych matematycznych specyfikacji systemu i przekształcaniu tych specyfikacji w program za pomocą metod matematycznych. Weryfikacja zgodności komponentów polega na wnioskowaniu matematycznych o ich zgodności ze specyfikacją.

Tworzenie z użyciem wielokrotnym (linie produkcyjne)

- W tym podejściu zakłada się istnienie dużej liczby komponentów zdatnych do ponownego użycia. Proces budowy systemu polega głównie na integrowaniu tych fragmentów, a nie na tworzeniu ich od początku.

Modele tworzenia systemów informatycznych

Model kaskadowy (czasem waterfall model)

- Długie cykle tworzenia oraz wdrażania.
- Sekwencyjne fazy akceptowane przez klienta.
- Programiści odseparowani od analityków.
- Słaba komunikacja między zespołami oraz klientem.

Proces Spiralny (Spiral Model)

- Krótkie okresy tworzenia oraz wdrażania.
- Przyspieszony proces iteracyjny.
- Elastyczne wspólne zespoły programistów i analityków.
- Powtarzalność faz.

Tworzenie ewolucyjne (Evolutionary Model)

- A jak to się ma do GUI?

Elementy RAD

Proces prototypowania

- *„Prototypowanie ewolucyjne ma wiele wspólnego z metodami błyskawicznego tworzenia programów użytkowych (Rapid Application Development, RAD)”* (*Ian Sommerville, „Inżynieria oprogramowania”, Rozdział 8, Prototypowanie oprogramowania)*

JAD: Joint Application Design

- Czasem nawet Joint Application Development

Zintegrowane narzędzia CASE

Generatory kodu

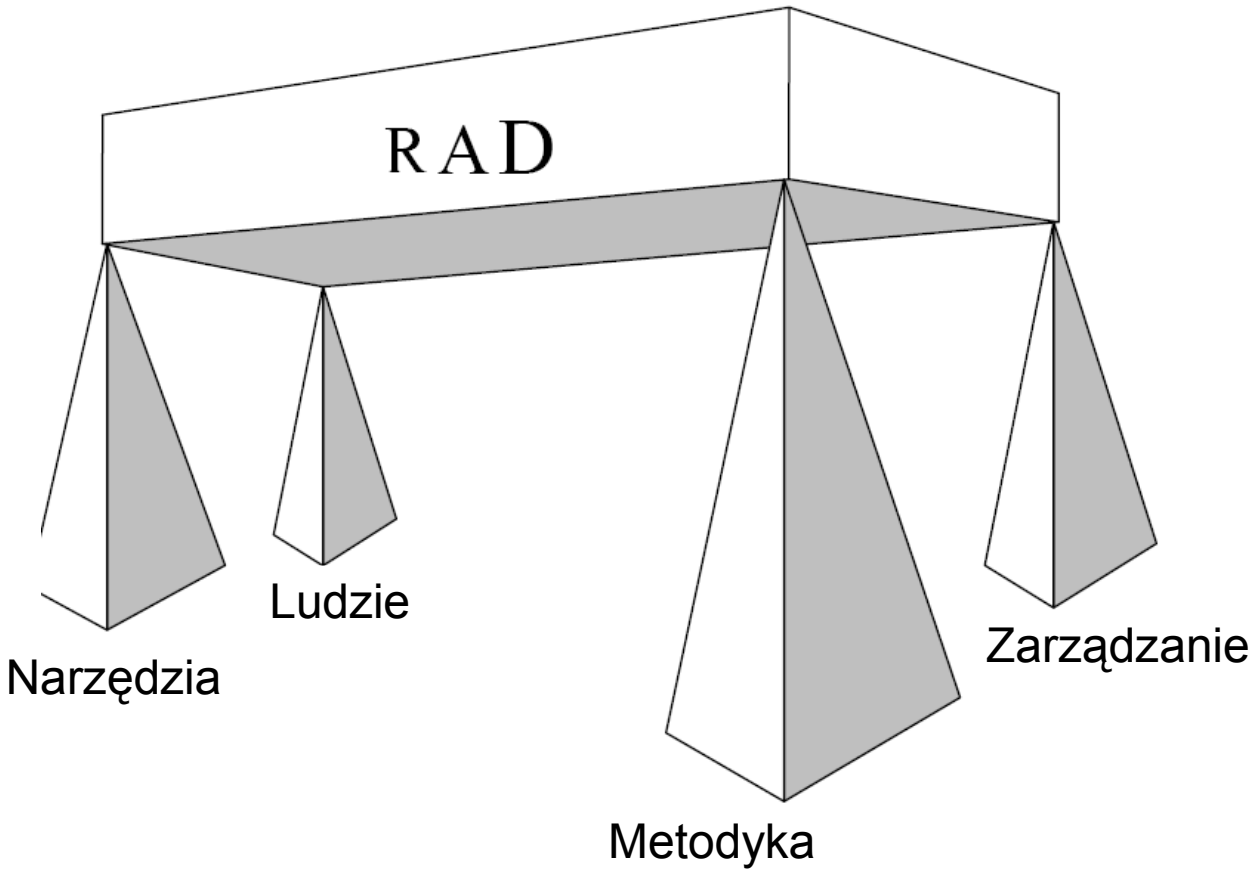
- narzędzia do projektowania GUI
- narzędzia do refaktoryzacji kodu
- narzędzia do kompilacji kodu i generowania wdrożeń

Podstawowe cechy RAD

Na podstawie: Ian Sommerville, „Inżynieria oprogramowania”, Rozdział 8.1

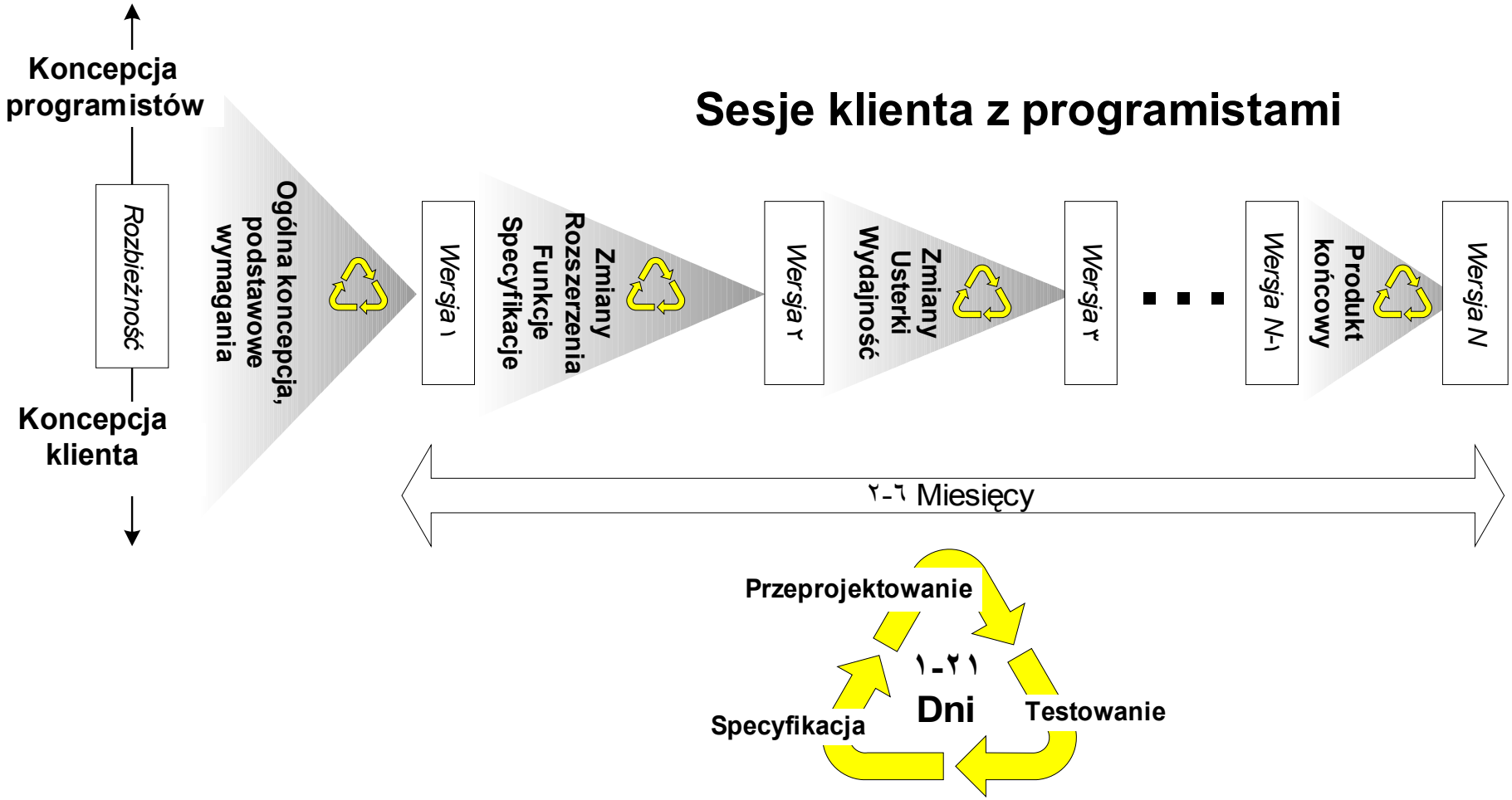
1. Procesy specyfikowania, projektowania i implementowania przeplatają się.
   - Nie ma szczegółowej specyfikacji systemu, a dokumentacja projektowa zwykle zależy od narzędzi użytych do implementacji systemu. Dokumentacja wymagań specyfikuje jedynie najważniejsze właściwości systemu.
2. System jest budowany w postaci ciągu przyrostów.
   - Użytkownicy i inni udziałowcy systemu są włączeni w projektowanie i ocenę każdego przyrostu. Mogą proponować zmiany w oprogramowaniu i nowe wymagania, które mają być zaimplementowane w późniejszej wersji systemu.
3. Stosuje się metody błyskawicznego tworzenia systemów.
   - Narzędzia CASE i języki czwartej generacji.
4. Interfejsy użytkownika budowane za pomocą interakcyjnego systemu wytwórczego,
   - który umożliwia szybkie tworzenie projektu interfejsu przez rysowanie i rozmieszczanie ikon.

Cztery filary RAD

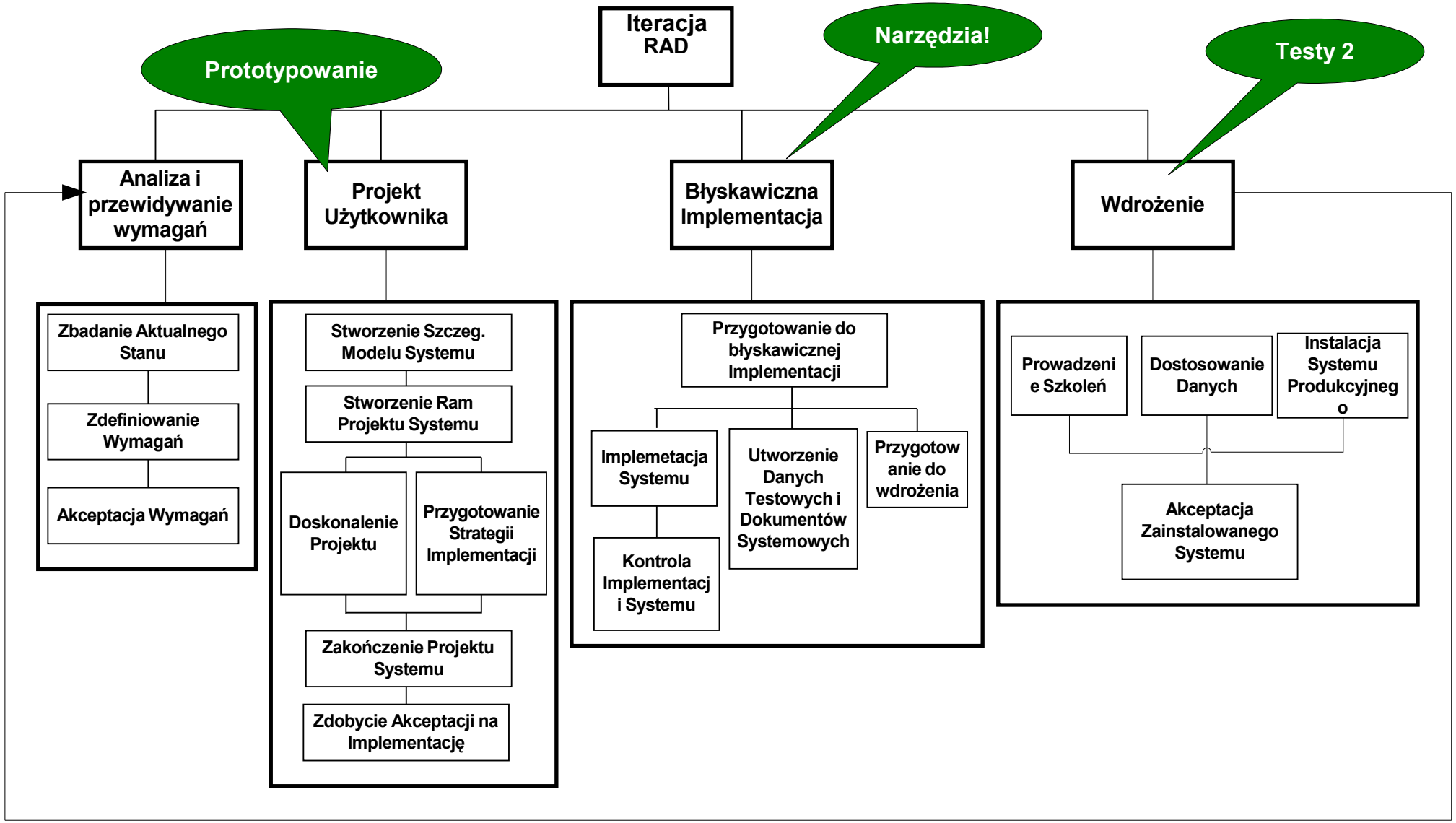


*Rysunek zmodyfikowany z: „**AN INTRODUCTION TO** RAPID APPLICATION DEVELOPMENT", Feb 2002, The Government of the Hong Kong Special Administrative Region*

Tworzenie ewolucyjne z pomocą prototypów



Etapy i Zadania RAD



Uczestnicy RAD

Koordynator spotkań, kontrolujący ich przebieg

Sekretarz (wpisywanie ustaleń do narzędzi CASE)

Zespół SWAT (Skilled With Advanced Tools)

Koordynator modelu

Administrator bazy danych

Zespół planujący spotkania robocze

Zespół projektowy użytkowników

Zespół wspomagający implementację

Zespół wdrożeniowy

http://www.radioluz.pwr.wroc.pl/upload/images/people_press_play.2007.people_press_play_(denmark).jpg

Założenia biznesowe

Timeboxing

- technika zarządzania stawiająca na pierwszym planie termin oddania systemu kosztem funkcjonalności.

możliwy dzięki:

- W pewnych sytuacjach 80% funkcjonalność systemu może być uzyskana w jednej piątej czasu potrzebnego do zrealizowania pełnej funkcjonalności.
- W pewnych sytuacjach, wymagania handlowe dla systemu mogą być w pełni zaspokojone nawet gdy niespełnione są niektóre wymagania operacyjne.
- W pewnych sytuacjach, może być uzyskana akceptacja systemu przy określonym minimum użytecznych funkcji a nie wszystkich.

Techniki stosowane w RAD

Modelowanie procesów biznesowych

- mało charakterystyczne, ale krytycznie wymagane aby RAD mógł zakończyć się sukcesem,

Budowanie prototypów

- podejście oparte na jak najwcześniejszym stworzeniu wersji, która może być zaprezentowana klientowi,

Iteracje

- podporządkowanie przyrostowemu tworzeniu aplikacji opartemu na udoskonalaniu wersji,

Timeboxing

- technika zarządzania stawiająca na pierwszym planie termin oddania systemu kosztem funkcjonalności,

Techniki stosowane w RAD

Spotkania robocze użytkowników

- JRP - Joint Requirements Planning,
- JAD - Joint Application Design

Kontrola przebiegu spotkań

- sformalizowana kontrola spotkań przez specjalnie wyznaczoną osobę,
- ta sama lub dodatkowa osoba dokumentuje spotkanie w narzędziu CASE,

Równoległa implementacja systemu

- wykorzystanie technik komponentowych, modularyzacja przez zespół SWAT (Skill With Advanced Tools).

Iteracyjne, ewolucyjne budowanie prototypów

Prototypy budowane przy bardzo bliskiej współpracy z przyszłymi użytkownikami

- 1. Projekt: JAD (Joint Application Design) – bliska współpraca przyszłych użytkowników oraz projektantów
- 2. Iteracje dopóki nie jest uzyskany pożądany efekt.
- 3. Stworzenie prototypu opartego na aktualnych wymaganiach.
- 4. Przegląd oraz ocena wewnętrzna prototypu.
- 5. Ocena prototypu przez klienta, sprecyzowanie dalszych wymagań.
- 6. Spotkanie robocze klienta z programistami.
- 7. Ocena czasu na realizację nowych wymagań i zmian.

Zalety RAD?

Krótki rozwój systemu. (3-6 miesięcy)

- Namacalne wyniki można uzyskać w bardzo krótkim czasie.

Użytkownicy mają do dyspozycji prototypy systemu.

- Nie tworzy się systemów nie spełniających wymagań.

Niższe koszty projektu, w dodatku łatwiejsze do kontroli.

Większe zaangażowanie twórców (użytkowników i projektantów).

Weryfikowane na bieżąco spełnienie kluczowych wymagań biznesowych.

- (80% funkcjonalności może być spełnione przy nakładzie 20%)

Czy jest druga strona medalu?

RAD często może doprowadzić do obniżenia jakości oprogramowania.

Tworzenie systemów „zapchaj dziura”.

Poświęcenie solidnej struktury systemu oraz metodologii na rzecz zyskania czasu.

Trudna kontrola postępu, za względu na brak podziału na etapy.

Większa liczba błędów z powodu „złego” kodu.

Zaufanie do dodatkowych, „obcych” komponentów.

Słabe ponowne wykorzystanie komponentów.

Słaba skalowalność produktu.

Brak dokumentacji.

http://www1.istockphoto.com

Czy RAD mogło być pigułką na problemy?

RAD nie jest lekarstwem na kryzys oprogramowania.

Nie skompensuje braku kompetencji autorów.

Nie wyeliminuje potrzeby zarządzania projektem.

Nie pomoże gdy zespół nie jest zdyscyplinowany pod względem kosztów oraz czasu realizacji.

Narzędzia w RAD

The Top 20 Tools Of All Time

http://uk.gizmodo.com/2006/03/23/the_top_20_tools_of_all_time.html

Wymagania dla środowisk RAD

Technologia „podnieś i upuść”

Narzędzia CASE (Computer Aided Software Engineering), m.in.:

- organizacja w odpowiedniej strukturze
- zintegrowane środowisko do manipulacji na danyc
- automatyczne generowanie dokumentacji

Obiektowa architektura języka

Powtórne używanie kodu

Rozszerzalność, elastyczność, skalowalność

Dostęp do wielu baz danych

Łatwy dostęp do serwisów wsparcia użytkowników

Popularność

Automatyczne generowanie kodu

Koordynacja prac wielu projektantów

Delphi jak się ma do RAD?

CodeGear from Borland

http://www.codegear.com/

Jak do tego pasuje Delphi?

Wielowarstwowe środowisko programistyczne

- Graficzny interfejs użytkownika
- Język Object Pascal
- Skalowalny interfejs do baz danych
- Bezpośredni dostęp do interfejsu API Windows
- Warstwa sieciowa (DCOM, MIDAS, CORBA, Internet)
- Zintegrowany debugger
- Biblioteki komponentów
- Możliwość tworzenia własnych zestawów narzędzi
  - (nie we wszystkich wersjach)
- Organizacja projektów

Krótka geneza Delphi

Chuck Jazdzewski (Chief Architect, Borland) - Button.Caption := OK;

Wcześniej: OWL, MFC (Turbo Vision)

Nowy język na podstawie Turbo Pascala

- Właściwości klas (Properties)
- Reprezentowanie obiektów przez wskaźniki
- Typ referencyjny (przekazywany do funkcji)
- Włączenie do kodu informacji o klasie i typach właściwości (RTTI – Runtime Type Information)
- Generowanie kodu maszynowego
- Bardzo prosty i naturalny do nauki język

Fenomenalnie szybki kompilator,

- dzięki nieskomplikowanej składni, która była projektowana również pod kątem wydajności procesu kompilacji

Ewolucja Delphi

Następca Turbo Pascala

Delphi 1 - elementy wizualne, procedury obsługi zdarzeń, biblioteki dll, bazy danych, 16 bitowy

Delphi 2 – 32 bitowy, zoptymalizowany kompilator, poszerzona biblioteka komponentów, udoskonalone mechanizmy dostępu do baz danych, dziedziczenie formularzy, technologia OLE

Delphi 3 – implementacja nowych technologii pod Windows: obiekty COM, kontrolki ActiveX, aplikacje serwerów WWW

Delphi 4 –wielowarstwowe aplikacje typu klient/serwer (MIDAS, CORBA czy DCOM)

Delphi 6 – udostępnie środowiska dla linuxa: Kylix

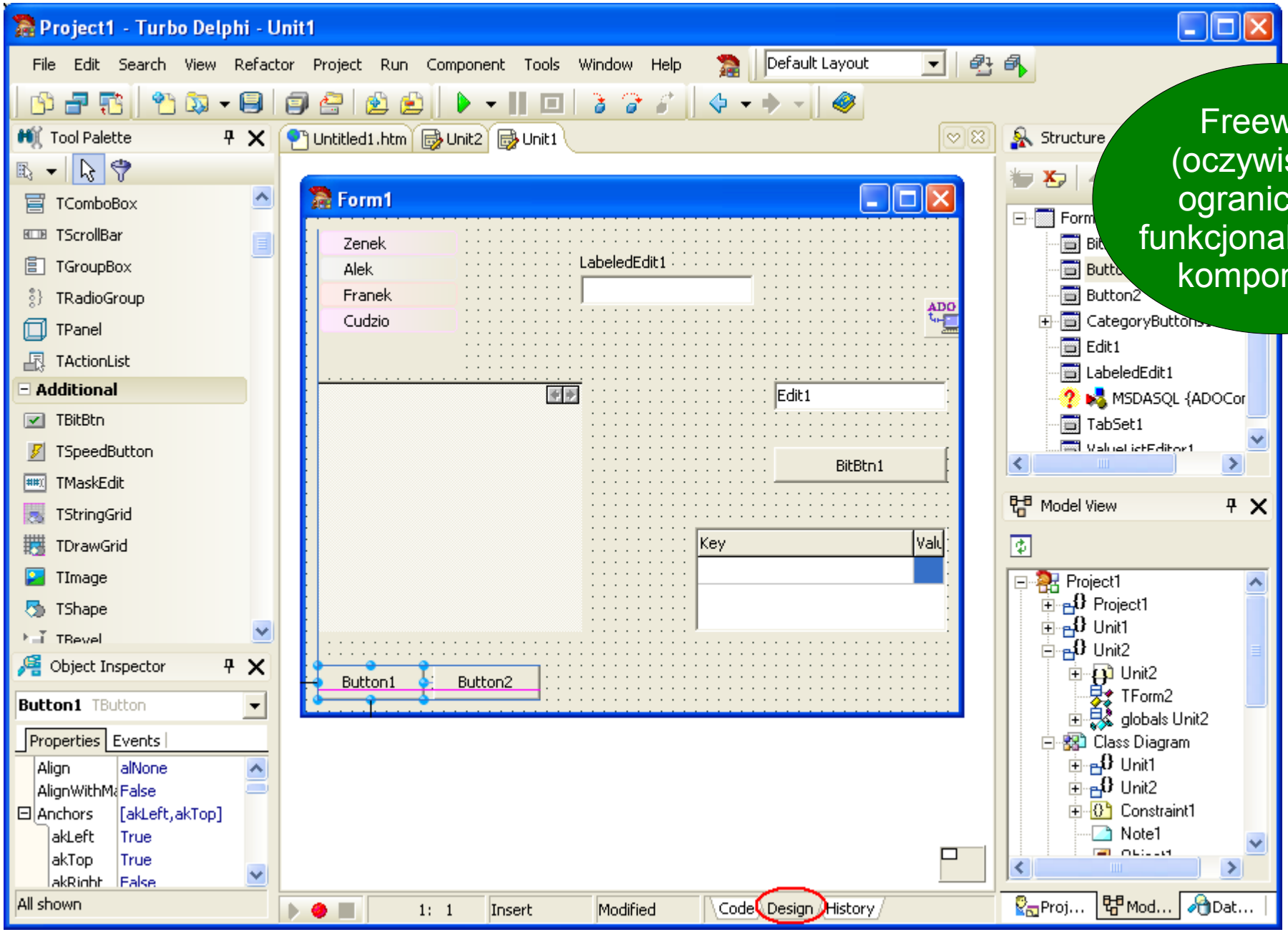
Delphi 7 – integracja z architekturą NET

Delphi 2005, 2006, 2007 for Win32

Turbo Delphi Explorer

Rok 2006,
Borland wystawia na sprzedaż dział rozwoju narzędzi programistycznych!

Turbo Delphi Explorer – Projekt wizualny



Freeware! (oczywiście z ograniczoną funkcjonalnością - komponenty)

Dziękuję za uwagę.

Chcemy być coraz lepsi!

Jeżeli coś cię zainteresowało napisz e-maila:

- robert@iem.pw.edu.pl

Jeżeli coś cię bardzo znudziło napisz e-maila:

- robert@iem.pw.edu.pl

Jeżeli zauważyłeś błąd napisz e-maila:

- robert@iem.pw.edu.pl