Overview

- Prototyping
- Agile Methods
- Using VB.Net to Prototype
- The Data Model as Design Tool
- What you’ll need to know
  - Forms, Variables, Classes, DB Connections etc.
  - When Prototyping goes bad
  - How to Manage the Prototyping Life-Cycle

Prototyping

- Prototype: ‘quick and dirty’ system that mimics the functionality of the final system
- Used to capture requirements
  - System features
  - Look and feel
- ‘Smoke and mirrors’
  - Prototype system usually doesn’t do any real processing.
  - Often developed with a ‘script’ in mind.

Prototyping

- Used in JAD sessions
  - Joint Application Development
  - User & development team sit in a room together, and a prototype is interactively developed
- Provides a quick means of determining and clarifying requirements
- Used as a technique for ‘Agile’ development methodologies

Agile Methods

- Hot new thing
- Rejection of the traditional, linear, waterfall model of the SDLC
- Characterised by:
  - Small teams
  - Incremental, almost chaotic development
  - Little up-front design and documentation
- A number of different methods within the ‘agile umbrella’
  - extreme Programming (XP)
  - Scrum
  - Adaptive Software Development
  - And others...
- Check out the manifesto:
  - http://agilemanifesto.org/
- More info at:
  - http://www.agilealliance.com
- Prototyping lies at the core of agile development

Requirements for Prototyping

- User availability
- Rapid development cycles
Prototyping in VB.Net

- Form design is key
- Database connections are a bit hard to do ‘on the fly’ though
  - Get the layout right with the users present
  - Do data connections later
- Choose the right controls
  - List box versus combo box
  - Check box versus radio button
- Colours and fonts
- Avoid coding with users present

The importance of the Data Model

- Chris talked about using the process model as the basis for design last week
- An alternative is to use the data model as a starting point
- Assume each table requires the following functionality:
  - Create
  - Read
  - Update
  - Delete
- Design screens to do each function for each entity, and you will have a skeleton system
- In general, you will use both models to finalise your design.

What you need to know

- Forms
- Variables, name spaces, classes
- Database connections

When prototyping goes bad

- Tendency to view the prototype as the system
  - May come as a surprise to the client that money and time is still required to build the real system
  - Sometimes you will want to turn the prototype into a fully working system, though
  - Need to be careful with code management

Code Management

- Version control systems
  - Need to be rigorous with:
    - Version numbering
    - Bug tracking
    - Use of test releases (alpha, beta etc)
- Bug tracking software
  - Bugzilla
  - Roll your own

Summary

- Prototyping is a quick and useful method for gathering requirements
  - Can be a useful design technique even if requirements have already been gathered
- Be careful of the pitfalls
  - Viewing the prototype as the final system
  - Code management
  - Project management