**Interface Building Tools**

**Features of Interface-Building Tools.**

User Interface Independence:

Separate interface design from internals

Enable multiple user interface strategies

Enable multiple platform support

Establish user interface architect role

Enforce standards

Methodology & Notation:

Develop design procedures

Find ways to talk about design

Create project management

Rapid Prototyping

Try out ideas very early

Test, revise, test, revise

Engage end users, managers, and others

Software Support

Increase productivity

Offer some constraint & consistency checks

Facilitate team approaches

Ease maintenance

**User interface mock-up tools**

Examples:

Paper and pencil

Word processors

Slide-show software

Macromedia Director, Flash mx, or Dreamweaver

Visual Editing

Microsoft Visual Studio

Borland JBuilder

**Finding the right tool is a trade-off between six main criteria:**

* 1. Part of the application built using the tool.
  2. Learning time
  3. Building time
  4. Methodology imposed or advised
  5. Communication with other subsystems
  6. Extensibility and modularity

**The windowing system layer**

Sometimes working at a low-level is required.

E.g., new platform

The while(true) main loop

**The GUI toolkit layer**

Widgets, such as windows, scroll bars, pull-down or pop-up menu, etc.

Difficult to use without an interface

**The application framework and specialized language layer**

Application frameworks are based on object-oriented programming

Can quickly build sophisticated interfaces

Require intensive learning

Specialized language layers lighten the programming burden

Tcl (and its toolkit Tk)

Perl/Tk

Python/Tk

Visual Basic

Java Script





