



The User Interface and Software Architecture

Len Bass

**Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890**

**Sponsored by the U.S.. Department of Defense
© 1997 by Carnegie Mellon University**



Overview

Five software qualities are important to the construction of the user interface:

- **usability**
- **buildability**
- **reusability**
- **modifiability**
- **performance**

Each quality has software architectural implications

Qualities interact



Usability

The ability for people to learn and use a user interface quickly, effectively and efficiently.

Architectural implications:

- **control model. If user has ability to input variety of commands at any time, must have structure to respond to unanticipated interrupts**
- **error handling. Centralized error handling and response enables consistency**
- **feedback. Feedback mechanisms both semantic and syntactic must be provided**



Buildability

System should be constructible within reasonable cost and schedule.

Architectural implications:

- **use of pre-existing components for user interface elements.**
- **use of components with common style for API**



Reusability

Using components from current system in future systems

Architectural implications:

- **keep interfaces uncomplicated**
- **choose functions computed by components to be general purpose**



Modifiability

Ability to easily make modifications to the user interface once it has been implemented.

Architectural implications:

- **separate user interface from application functionality**
- **within user interface, allocate functionality based on likely pattern of changes**



Performance

Provide adequate response to user input.

Architectural implications:

- **reduced number of interfaces**
- **reduced communication paths**
- **caching data**



Interactions Among Qualities

Interactions among qualities prevent achieving all of any one quality. For example:

- **performance is counter to other qualities**
 - **reducing interfaces increases specificity of components and hurts reuse and modifiability**
- **buildability by using pre-existing components hurts modifiability and usability**
- **reuse hurts buildability by requiring more components (general and tailored)**
- **usability hurts reuse by including application specific feedback and error handling.**



Summary

Many disputes over design are because of disagreements over priority of different qualities.

Can never achieve optimum result for all qualities simultaneously. Must have trade offs.

Understanding qualities affected and interactions among them at architectural level provides structure for resolving disagreements over priorities.