Designing software with users and testers in mind

Alexandra Schladebeck

BREDEX GmbH
Introductions

Usability

two birds with one stone

Testability
Usability and testability

Usability

Visibility of system status
User control and freedom
Consistency and standards
Flexibility and efficiency
Recognition rather than recall
Recognize, diagnose, recover from errors

Testability

Observability
Controllability
Understandability
Automatability
Simplicity
### Visibility of system status

**Usability**
- Keep users informed
- Feedback
- Reasonable time

**Testability**
- Observe results
- Automated checks

### Usability
- Visibility of system status
- User control and freedom
- Consistency and standards
- Flexibility and efficiency
- Recognition rather than recall
- Recognize, diagnose, recover from errors

### Testability
- Observability
- Controllability
- Understandability
- Automatability
- Simplicity
1 Making progress

- **Usability**: - -
  Is it broken?
  Should I restart?

- **Testability**: - -
  When is it done?
1 Making progress

- Static wait

- Synchronize
  - Static wait

- Trigger action
  - Select
1 Making progress

- **Usability: -**
  - Is it broken?
  - Will it ever be finished?

- **Testability: 0**
  - When is it done?
1 Making progress

- Trigger action
  - Press save

- Synchronize
  - Wait for cursor ?
  - Static wait
1 Making progress

- **Usability**: +
  It’s not broken!
  How long will it take?

- **Testability**: ++
  Orient to dialog
1 Making progress

- Trigger action
  - Choose "connect"

- Synchronize
  - (Wait for window)
  - Wait for window to close
1 Making progress

- **Usability: ++**
  Feedback about time

- **Testability: ++**
  Just as good 😊
2 Looking for clues

- **Usability:** ++
  - Reaction shown
  - Multiple signs

- **Testability:** ++
  - Test can orient to state
  - Sanity check for success
2 Looking for clues

- Click save
- Wait for save to become disabled
- Check no * on tab name

Save editor
## Usability
- Emergency exits

## Testability
- Control state of component

### Usability
- Visibility of system status
- User control and freedom
- Consistency and standards
- Flexibility and efficiency
- Recognition rather than recall
- Recognize, diagnose, recover from errors

### Testability
- Observability
- Controllability
- Understandability
- Automatability
- Simplicity
3 Offer a way out

**Usability:** --
No chance to cancel
Confirmation via X?

**Testability:** --
Access to X
Save / cancel use case?
3 Offer a way out

- **Usability: ++**
  - Clear choice
  - Clear effects

- **Testability: ++**
  - Orientation to buttons
  - Access to buttons
4 I’ve changed my mind

- **Usability:** --
  Cannot cancel

- **Testability:** --
  Cannot test cancellation
4 I’ve changed my mind

- **Usability**: ++
  User in control

- **Testability**: ++
  Cancellation use case
5 Are you sure?
5 Are you sure?

- **Usability: ++**
  - Option to check/cancel
  - Explicit action

- **Testability: ++**
  - Orient to dialog closure
  - Reuse delete module
Consistency and standards

- **Usability**
  - Follow conventions
  - Use same meaning

- **Testability**
  - Self-explanatory
  - Lends itself to automation

### Usability

- Visibility of system status
- User control and freedom
- Consistency and standards
- Flexibility and efficiency
- Recognition rather than recall
- Recognize, diagnose, recover from errors

### Testability

- Observability
- Controllability
- Understandability
- Automatability
- Simplicity
6 I know where I am?

**Delete node from tree with confirmation**
- Select node
- Select delete from context menu
- Wait for confirmation dialog
- Click OK
- Wait for confirmation dialog to close
- Check node no longer exists

**Usability: --**
When does confirm come?

**Testability: --**
Two modules

**Delete node from tree w/o confirmation**
- Select node
- Select delete from context menu
- Check node no longer exists
Delete node from tree

- Select node
- Select delete from context menu
  - Wait for confirmation dialog
  - Click OK
- Wait for confirmation dialog to close
- Check node no longer exists

Usability: ++
Always confirmed

Testability: ++
More reuse in tests
**Usability**
- Accelerators
- Support preference
- Accessibility

**Testability**
- Access to all functions

<table>
<thead>
<tr>
<th>Usability</th>
<th>Testability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visibility of system status</td>
<td>Observability</td>
</tr>
<tr>
<td>User control and freedom</td>
<td>Controllability</td>
</tr>
<tr>
<td>Consistency and standards</td>
<td>Understandability</td>
</tr>
<tr>
<td>Flexibility and efficiency</td>
<td>Automatability</td>
</tr>
<tr>
<td>Recognition rather than recall</td>
<td>Simplicity</td>
</tr>
<tr>
<td>Recognize, diagnose, recover from errors</td>
<td></td>
</tr>
</tbody>
</table>
7 My way or the highway

- **Usability:** --
  Only one option
  Inaccessible

- **Testability:** --
  Custom controls
  Only one option
My way or the highway

- **Usability:** ++
  - Preference support
  - More accessible

- **Testability:** ++
  - Increased access
  - Increased reusability
### Possible alternatives

<table>
<thead>
<tr>
<th>Mouse</th>
<th>Keyboard</th>
<th>Gesture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu</td>
<td>Shortcut</td>
<td>Swipe</td>
</tr>
<tr>
<td>Context menu</td>
<td>Mnemonic</td>
<td>...</td>
</tr>
<tr>
<td>Toolbar</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Recognition rather than recall

- **Usability**
  - Minimize memory load
  - Visibility

- **Testability**
  - Fewer clicks
  - Less repetition (maintenance)
  - Understandable
  - Simple

### Usabililty
- Visibility of system status
- User control and freedom
- Consistency and standards
- Flexibility and efficiency
- Recognition rather than recall
- Recognize, diagnose, recover from errors

### Testability
- Observability
- Controllability
- Understandability
- Automatability
- Simplicity
# Usability
- Indicate problem
- Suggest solution
- Less frustration

# Testability
- Intermediate and final results

## Usability
- Visibility of system status
- User control and freedom
- Consistency and standards
- Flexibility and efficiency
- Recognition rather than recall
- Recognize, diagnose, recover from errors

## Testability
- Observability
- Controllability
- Understandability
- Automatability
- Simplicity
Recognize, diagnose, recover from errors

- Show nothing
  - Usability: --
  - Testability: --

- Show error dialog
  - Usability: 0
  - Testability: +

- Don’t allow confirm
  - Usability: +
  - Testability: +

- Inline corrections
  - Usability: ++
  - Testability: ++

- Errors impossible

Error handling
Smells

- Writing the docs makes you die a little inside
- Support questions driving you nuts
- You notice how many clicks you have to automate
- The GUI offers no clues
- Static waits are your best friend
- Writing different modules (Test Cases) for same activity
Top three
Questions?
References

- Shneidermann’s 8 Golden Rules of Interface Design
  http://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html

- Nielsen’s 10 Usability Heuristics
  http://en.wikipedia.org/wiki/Heuristic_evaluation

- Testability
  http://en.wikipedia.org/wiki/Software_testability
Give Feedback on the Sessions

1. Sign In: www.eclipsecon.org

2. Select Session Evaluate

3. Vote +1 0 -1