

# User Interaction Design

## User Centered Design, Usability Principles, Design Process, Identifying the User, Tools, Usability Tests

Usability is an intangible software feature - more noticeable by its absence than its presence in an application. This course tells you how to design it into your products. User interaction design is a branch of software engineering – and needs to be approached in an organized manner. We need to get to know the user, to discover the user interaction requirements, to build the app, and then to test that we have done it correctly. This is both similar and different to how we would code up an algorithm or implement a database schema. Every workitem on a software dev team’s project plan should be examined to see if it enhances usability of the app -

if not, it needs to be reconsidered. There is a need to collect as much information as possible from the user – via product demonstrations, usability testing, and after launch, analyzing the helpdesk queries. Just like advertising, 50% of software features are never used – you will need to find out which, and more importantly why – perhaps the unused features are too difficult, perhaps the user does not know they exist (inaccessible), or perhaps they are simply not needed. Designers and developers will benefit from attending this course by gaining an increasing awareness of the processes used to create incredible user-friendly apps.

<b>Contents of One-Day Training Course</b>	
<p><b>Target Audience</b> This course is aimed at software designers and senior developers who need to create highly intuitive user interfaces</p> <p><b>Prerequisites</b> Experience of programming graphical user interfaces is needed along with an appreciation of usability issues</p>	<p><b>User-Centric Design</b> Placing the user at the center of the software design process Every developer needs to think of the user experience – not just those who directly create the user interface</p> <p><b>Usability Design Principles</b> Terminology and metaphors Consistency Task invocation and navigation Functionality discovery Continuous feedback Controllability Selection and activation Validation of user input</p> <p><b>Concepts</b> User – application communication Norman’s Model From CLI to GUI to CBI Command-Based Interface used from multiple sources (GUI, wizards, macros, Automation, CBT/Help, undo/redo)</p> <p><b>Usability Design Process</b> Identifying the user Task analysis Story-boarding Prototyping Usability testing An iterative cycle</p> <p><b>Direct Manipulation</b> Direct interaction In-place editing Drag and drop</p>
	<p><b>The look of the application</b> Font, color, etc. Use of certain controls Layout of forms Task-centric design Icons and cursors Benefits of a “minimalist” interface</p> <p><b>Types of GUI</b> OOUI MUI Handling complexity Handling large data sets</p> <p><b>Usability for different apps</b> Database (transactions, locking, tables) Graphics (dirty bit, perspective, selection) Components (object display, activation) Networking (node selection, time delays)</p> <p><b>User Assistance Tools</b> Help system/online mentoring wizard Context sensitive computer based training</p> <p><b>Error Avoidance</b> Why errors happen Engineering user errors out of applications Users never make errors-only designers do</p> <p><b>Usability Testing</b> The “five minute” user test Collecting information from users User interaction engineering Making it part of the development project and what internal doc/models are needed.</p> <p><b>Usability Engineering Project</b> Complete walk through of how to design user interaction for a complex project</p>



<http://www.clipcode.net/training>

To arrange an on-site presentation anywhere in Europe, please email [training@clipcode.com](mailto:training@clipcode.com)